

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: September 30, 2002

Application Number: Patent Application
No. 2002-285046
[ST.10/C]: [JP2002-285046]

Applicant(s): FUJITSU LIMITED

January 24, 2003

Commissioner,
Japan Patent Office Shinichiro OTA

Certificate No. P2003-3001441

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 9月30日

出 願 番 号

Application Number:

特願2002-285046

[ST.10/C]:

[JP2002-285046]

出 願 人

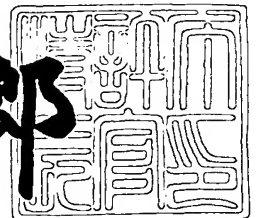
Applicant(s):

富士通株式会社

2003年 1月24日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3001441

【書類名】 特許願

【整理番号】 0251706

【提出日】 平成14年 9月30日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/45

【発明の名称】 プログラムを最適化する制御をコンピュータに行わせる
コンピュータ・プログラム及び、プログラムの最適化を
行う最適化処理装置

【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通
株式会社内

【氏名】 吉良 正規

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100074099

【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F

【弁理士】

【氏名又は名称】 大菅 義之

【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾7-25-28-503

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラムを最適化する制御をコンピュータに行わせるコンピュータ・プログラム及び、プログラムの最適化を行う最適化処理装置

【特許請求の範囲】

【請求項 1】 プログラムを最適化する制御をコンピュータに行わせるコンピュータ・プログラムであって、

前記プログラムからデータ項目を抽出し、

前記データ項目を前記コンピュータに備えられたメモリ上にレイアウトし、

抽出された前記データ項目から、定義されたが使用されない未使用データ項目を抽出し、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をもつデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新たなデータ項目に集約することが可能か否か判定し、

前記判定結果に基づいて、前記複数のデータ項目を前記1つの新たなデータ項目に集約したプログラムを出力する、

ことを含む制御を前記コンピュータに行わせることを特徴とするコンピュータ・プログラム。

【請求項 2】 集約される前記複数の未使用データ項目は、前記メモリ上において互いに隣接する領域にレイアウトされている、

ことを特徴とする付記 1 に記載のコンピュータ・プログラム。

【請求項 3】 集約される前記複数の未使用データ項目は、共に階層構造をもつ他のデータ項目を構成するデータ項目であって、且つ、前記階層構造におけるレベルが互いに同じである、

ことを特徴とする請求項 2 に記載のコンピュータ・プログラム。

【請求項 4】 集約される前記複数の未使用データ項目は、階層構造をもつデータ項目と当該データ項目を構成するデータ項目であり、且つ、前記階層構造をもつデータ項目は1つのデータ項目によって構成される、

ことを特徴とする請求項 1 に記載のコンピュータ・プログラム。

【請求項 5】 プログラムを最適化する最適化処理装置であって、

前記プログラムからデータ項目を抽出するデータ項目抽出手段と、
抽出された前記データ項目をメモリ上にレイアウトするレイアウト手段と、
抽出された前記データ項目から、定義されたが使用されない未使用データ項目
を抽出する未使用データ項目抽出手段と、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をも
つデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新
たなデータ項目に集約することが可能か否か判定する集約判定手段と、

前記判定結果に基づいて、前記複数のデータ項目を前記1つの新たなデータ項
目に集約した前記プログラムを出力するデータ項目集約手段と、
を備えることを特徴とする最適化処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プログラムを最適化するための技術に関し、特に、プログラム中の
未使用データ項目を集約・削減する技術に関する。

【0002】

【従来の技術】

プログラム中には、変数等のデータ項目が定義されることが多い。しかし、定
義された全てのデータ項目が、プログラム中において使用（参照）されていると
は限らない。以下において、定義されたが、プログラム中で使用されないデータ
項目を未使用データ項目という。コンパイル時やプログラム解析時等において、
データ項目辞書が作成されるが、データ項目辞書に未使用データ項目が含まれて
いる場合、その分、処理に要するメモリ容量が余分に必要になる。未使用データ
項目はプログラム中で使用されないため、そのデータ項目を定義するコードは必
要ではない。そこで、コンパイラには、メモリ資源を有効に利用できるようにす
るために、未使用データ項目を定義するコードを削除する機能が与えられている
ことが多い。

【0003】

以下、未使用データ項目を定義するコードを削除する機能について、コンパイ

ラを例として説明する。なお、実際のコンパイラは、このような機能以外に、単純代入の除去、定数計算の最適化等の機能を有することが多いが、説明を簡単にするために、コンパイラは未使用データ項目を定義するコードの削除以外の最適化を行わないと仮定する。

【 0 0 0 4 】

図 3 4 に、COBOL (COmmonBusiness Oriented Language) で記述されたソースプログラムの一例を示す。このソースプログラムでは、130 行目から 170 行目までにおいて定義された変数である IREC01、IDATA11、IDATA12、IDATA13 及び IDATA14 は、300 行以降の処理部 (PROCEDURE DIVISION) で参照されていないため、未使用データ項目である。同様に、210 行目、220 行目、及び 240 行目から 270 行目までにおいて定義された変数である WDATA12、WDATA13、WREC02、WDATA21、WDATA22、及び WDATA71 も未使用データ項目である。

【 0 0 0 5 】

図 3 4 に示すソースプログラムから従来技術を用いて未使用データ項目を定義するコード除去すると、図 3 5 に示すようなソースプログラムが得られる。図 3 5 において、240 行目から 270 行目までのコードが除去されている。

【 0 0 0 6 】

図 3 6 に、C で記述されたソースプログラムの一例を示す。図 3 6 に示すソースプログラムは、図 3 4 に示す COBOL で記述されたソースプログラムとほぼ同じ内容をもつ。図 3 7 に、従来技術を用いて図 3 6 に示すソースプログラムから未使用データ項目を定義するコードを除去した結果を示す。

【 0 0 0 7 】

不要な変数の削除に関する技術として、プログラム中に型宣言している変数が処理記述で使用されたか否かを確認し、使用されていないことが分かった場合、エラーメッセージを出力装置に出力させる事により、プログラム中の無駄な型宣言をユーザに通知すると共に、無駄に型宣言された変数がオブジェクトファイル中に生成されないようにする機能を有するコンパイラがある。コンパイラにより、コンパイル後のプログラム中には無駄な型宣言が含まれないようになる (例えば、特許文献 1 参照)。

【 0 0 0 8 】

【特許文献 1】

特開平 7 - 2 3 9 7 8 8 号公報（段落 0 0 1 0 から段落 0 0 1 5、図 1
）

【 0 0 0 9 】

【発明が解決しようとする課題】

しかし、上記の従来技術によれば、各データ項目に割り振られたメモリ上のアドレスがずれることを防ぐために、以下のような未使用データ項目を除去できないという問題があった。

【 0 0 1 0 】

- ・単純なデータ項目
- ・レコード（階層構造を有するデータ項目）の少なくとも一部を構成するデータ項目

図 3 4 及び 3 5 に示す COBOL で記述されたソースプログラムを用いて、上記問題についてより具体的に説明する。図 3 4 に示すソースプログラム中の 1 3 0 行目から 1 7 0 行目まで、2 1 0 行目、及び 2 2 0 行目において定義された IREC01、IDATA11、IDATA12、IDATA13、IDATA14、WDATA12 及び WDATA13 は、未使用データ項目であるにもかかわらず、図 3 5 に示すソースプログラムにおいて、これらの未使用データ項目は除去されない。未使用データ項目 IREC01 は、7 0 行目の SELECT 文で IFILE01 が使用されており、未使用データ項目 IDATA11、IDATA12、IDATA13 及び IDATA14 は、レコード IREC01 を構成するデータ項目であり、また、未使用データ項目 WDATA12 及び WDATA13 は、レコード WREC01 を構成するデータ項目であるためである。

【 0 0 1 1 】

図 3 6 及び 3 7 に示す C で記述されたソースプログラムから、COBOL だけでなく他の言語であっても、同様の未使用データ項目を除去できないということが分かる。

【 0 0 1 2 】

しかしながら、コンピュータにおけるメモリ領域は無限にあるわけではない。

ソースプログラムの除去できない未使用データ項目がそのまま処理されてしまうと、コンパイル時やデータ項目辞書の作成時にメモリ領域が不足して処理できないことがありえる。

【 0 0 1 3 】

上述のように、ソースプログラムをコンパイルしたり、ソースプログラムのデータ項目辞書を作成したりする場合に必要なメモリ量を低減するためには、ソースプログラムに含まれる未使用データ項目の数が少ない事が望ましい。

【 0 0 1 4 】

以上の問題に鑑み、従来の技術よりも効果的に未使用データ項目の数を削減することを可能とし、これにより、コンパイル時やデータ項目辞書作成時に必要となるメモリ量を低減するようにプログラムを最適化することを可能とすることが、本発明の解決しようとする課題である。

【 0 0 1 5 】

【課題を解決するための手段】

上記問題を解決するために、本発明の1態様によれば、プログラムを最適化する最適化処理装置において、上記プログラムからデータ項目を抽出するデータ項目抽出手段と、抽出された上記データ項目をメモリ上にレイアウトするレイアウト手段と、抽出された上記データ項目から、定義されたが使用されない未使用データ項目を抽出する未使用データ項目抽出手段と、上記レイアウト結果に基づいて、上記未使用データ項目のうち、階層構造をもつデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新たなデータ項目に集約することが可能か否か判定する集約判定手段と、上記判定結果に基づいて、上記複数のデータ項目を前記新たなデータ項目に集約した上記プログラムを出力するデータ項目集約手段と、を備えるように構成する。

【 0 0 1 6 】

従来、階層構造をもつデータ項目の少なくとも一部を構成する未使用データ項目を削除する事はできなかった。しかし、上記構成によってメモリ上のレイアウトに基づいてこのような未使用データ項目のうちの複数の未使用データ項目を1

つに集約することにより、従来技術よりも効果的に未使用データ項目の数を削減することが可能となる。延いては、コンパイル時やデータ項目辞書作成時に必要となるメモリ量を低減するようにプログラムを最適化することが可能となる。

【0017】

ここで、集約される上記複数の未使用データ項目は、上記メモリ上において互いに隣接する領域にレイアウトされていることとしてもよい。

データ項目を集約可能であると判定する場合には、2つの場合がある。まず、複数の未使用データ項目が、共に階層構造をもつ他のデータ項目を構成するデータ項目であって、且つ、上記階層構造における階層レベルが互いに同じである場合、上記集約判定手段はその複数の未使用データ項目を集約可能であると判定する。また、複数の未使用データ項目が、階層構造をもつデータ項目と当該データ項目を構成するデータ項目であり、且つ、上記階層構造をもつデータ項目は1つのデータ項目によってのみ構成される場合も、上記集約判定手段はその複数の未使用データ項目を集約可能であると判定する。

【0018】

また、上記データ項目集約手段は、複数の未使用データ項目を宣言するコードを前記プログラムから削除し、1つの新たなデータ項目を宣言するコードを追加することとしても良い。プログラムから不要な定義コードを削除することにより、プログラムを最適化することが可能となる。

【0019】

さらに、また、上記データ項目集約手段は、集約される上記複数の未使用データ項目の項目長の和を算出し、算出された上記項目長の和に基づいて上記新たなデータ項目の項目長を設定することとしてもよい。これにより、集約によって、集約前後でデータ項目のメモリ上のアドレスが変更されることを防ぐことが可能となる。

【0020】

上記データ項目の集約において、宣言される新たなデータ項目のデータ型の設定には、複数の考え方がありうる。まず、上記データ項目集約手段は、集約される上記複数の未使用データ項目のデータ型が全て同じである場合、集約される上

記複数の未使用データ項目のデータ型と同じデータ型として、上記新たなデータ項目のデータ型を設定することとしてもよい。これにより、集約によって、集約されたデータ項目のデータ型が集約前のデータ項目のデータ型と異なるデータ型となることを防ぐことが可能となる。

【 0 0 2 1 】

また、上記データ項目集約手段は、上記新たなデータ項目のデータ型を、記憶域サイズが最小となるデータ型として設定することとしてもよい。記憶域サイズが最小となるデータ型を、新たな項目のデータ型として決定しておくことにより、データ型の設定を容易にすることが可能となる。データ型の設定に困難が生じる場合として、例えば、集約される複数の未使用データ項目のデータ型が互いに異なる場合が挙げられる。また、例えば、新たなデータ項目の項目長を集約前の複数の未使用データ項目の項目長の和と等しくするためには、集約前のデータ項目のデータ型と異なるデータ型を新たなデータ項目に設定する必要がある場合が挙げられる。後者の場合の具体例として、例えば、短精度整数データのような精度が指定された数値データとして宣言された未使用データ項目を集約する場合が挙げられる。

【 0 0 2 2 】

また、上記データ項目の集約において、宣言される新たなデータ項目の項目名の設定にも、複数の考え方がある。例えば、上記データ項目集約手段は、上記新たなデータ項目の項目名を無名として設定することとしてもよい。また、上記データ項目集約手段は、上記新たなデータ項目の項目名を、集約される上記複数の未使用データ項目のうちの何れか1つに基づいて設定することとしてもよい。また、上記データ項目集約手段は、最適化処理装置のユーザの指定に基づいて上記新たなデータ項目の項目名を設定することとしてもよい。

【 0 0 2 3 】

また、本発明の別の1態様によれば、コンピュータが、プログラムを最適化する最適化方法において、上記プログラムからデータ項目を抽出し、上記データ項目を上記コンピュータに備えられたメモリ上にレイアウトし、抽出された上記データ項目から、定義されたが使用されない未使用データ項目を抽出し、上記レイ

アウト結果に基づいて、上記未使用データ項目のうち、階層構造をもつデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新たなデータ項目に集約することが可能か否か判定し、上記判定結果に基づいて、上記複数のデータ項目を上記新たなデータ項目に集約したプログラムを出力する、ことを含むように構成する。このプログラム最適化方法によっても、上記の最適化処理装置と同様の作用・効果を得ることができるため、上記課題を解決することが可能である。

【 0 0 2 4 】

また、上記プログラム最適化方法において行われる手順をコンピュータに実行させるコンピュータ・プログラムも、上記コンピュータ・プログラムをコンピュータに備えられたメモリに一旦格納させ、そのコンピュータにそのメモリから上記コンピュータ・プログラムを読み出させて実行させることにより、上記課題を解決する事が可能である。

【 0 0 2 5 】

また、上記コンピュータ・プログラムを記録したコンピュータ読み取り可能な記録媒体から、そのプログラムをコンピュータに読み出させて、上記プログラムにロードさせ、その後、上記と同様ににしてそのコンピュータにそのコンピュータ・プログラムを実行させることによっても、上記課題を解決することができる。

【 0 0 2 6 】

【発明の実施の形態】

以下、本発明の実施の形態について図面を用いて説明する。なお、同じ装置等には同じ参照番号をつけ、説明を省略する。

【 0 0 2 7 】

図1に、本発明に係わる最適化処理装置1の構成を示す。最適化処理装置1は、ソースプログラムに含まれる未使用データ項目群を集約することによって、ソースプログラムに含まれるデータ項目数を削減するようにソースプログラムを最適化する。図1に示すように、最適化処理装置1は、データ項目抽出部2、データ項目辞書作成部3、レイアウト作成部4、未使用データ項目抽出部5、集約判

定部 6、判定結果通知部 7、集約指示受付部 8、データ項目集約部 9、入出力部 10、プログラム格納部 11、データ項目辞書格納部 12 及び未判定データ項目リスト格納部 13 を備える。

【 0 0 2 8 】

データ項目抽出部 2 は、プログラム格納部 11 からソースプログラムを取得し、そのソースプログラムからデータ項目を抽出する。データ項目辞書作成部 3 は、抽出されたデータ項目の内容及びデータ項目間の関係を示すデータ項目辞書を作成し、データ項目辞書格納部 12 に書き込む。レイアウト部 4 は、抽出されたデータ項目をメモリ（不図示）上に配置（レイアウト）する。未使用データ項目抽出部 5 は、抽出されたデータ項目のうち、未使用データ項目を抽出する。集約判定部 6 は、メモリ上でのレイアウトに基づいて、抽出された未使用データ項目から、複数の未使用データ項目からなる組を 1 つのデータ項目に集約できるか否か判定する。判定結果通知部 7 は、入出力部 10 を介して集約判定部 6 によって集約できると判定された複数の未使用データ項目を最適化処理装置 1 のユーザに通知する。集約指示受付部 8 は、入出力部 10 を介してユーザからデータ項目を集約すべきか否かを示す指示を受け付ける。データ項目集約部 9 は、ユーザからの指示に基づいてデータ項目を集約し、入出力部 10 を介して集約結果をユーザに通知する。入出力部 10 は、ユーザと最適化装置 1 の間における情報のやり取りに用いられる。

【 0 0 2 9 】

プログラム格納部 11 は、入出力部 10 を介して入力されたソースプログラム、及び最適化後のソースプログラムを格納する。データ項目辞書格納部 12 は、データ項目辞書作成部 3 によって作成されたデータ項目辞書を格納する。未判定データ項目リスト格納部 13 は、未判定データ項目リストを格納する。未判定データ項目リストは、未使用データ項目の抽出結果及びレイアウト結果に基づいて集約判定部 6 によって作成され、データ項目を集約することができるか否か判定する際に、判定対象となるデータ項目及び判定結果を管理するために利用される。

【 0 0 3 0 】

次に、図 2 から図 2 7 を用いて、最適化処理装置 1 によって行われる処理について説明する。まず、図 2 を用いて、プログラムの最適化処理の大まかな流れについて説明する。

【 0 0 3 1 】

図 2 に示すように、まず、最適化処理装置 1 は、プログラム格納部 1 1 から最適化されるべきソースプログラムを取り出して、そのソースプログラムにおけるデータフローを解析し、データ項目の定義及び参照情報を収集する（S 1）。なお、データフローの解析は、コンパイラ等における一般的なプログラムの最適化手法である。データフロー解析の手順は以下のとおりである。

【 0 0 3 2 】

1) データ項目抽出部 2 は、前処理実行後の翻訳テキストに対して、データ項目の宣言情報を収集する。翻訳テキストの例として、COBOL の COPY 文、C 又は C++ の include 文などを展開した後のソースプログラムが挙げられる。データ項目辞書について詳しくは後述する。

【 0 0 3 3 】

2) レイアウト部 4 は、階層構造をもつデータ項目について、いかなるデータ項目の構成要素となっていない項目（最外のデータ項目）を基準として、データ項目のレイアウト上の位置を示す相対変位を計算し、それらのデータ項目間の重なり情報を収集する。このとき、レイアウト部 4 は、ソースプログラムを記述するプログラム言語に応じたデータ項目間の間隔の補正を考慮する。データ項目間の間隔の補正として、例えば、COBOL における遊びバイト又遊びビット、C 又は C++ における語境界等が挙げられる。レイアウトについてより詳しくは後述する。

【 0 0 3 4 】

3) データ項目辞書作成部 3 は、宣言情報に基づいてデータ項目辞書を作成し、さらに、データ項目の定義・参照情報を収集して、データ項目辞書にその定義・参照情報を設定する。例えば、図 3 に示すソースプログラムにおいて、レコード WREC01 を構成するデータ項目の 1 つである WDATA11 が参照されていた場合、データ項目辞書作成部 3 は、データ項目 WDATA11 とレコード WREC01 に対する定義・参照情報をデータ項目辞書に設定する。

【 0 0 3 5 】

次に、未使用データ項目抽出部 5 は、抽出されたデータ項目のうち、階層構造をもつデータ項目（レコード）を抽出する（S 2）。例えば、図 3 に示すソースプログラムを最適化する場合、未使用データ項目抽出部 5 は、階層構造をもつデータ項目として、IREC01、WREC01 及び WREC02 を抽出する。

【 0 0 3 6 】

未使用データ項目抽出部 5 が階層構造をもつデータ項目を抽出できなかった、つまり、最適化処理の対象となるソースプログラム中に階層構造をもつデータ項目が存在しない場合（S 3 : N o）、処理を終了する。

【 0 0 3 7 】

階層構造をもつデータ項目を抽出できた場合（S 3 : Y e s）、さらに、未使用データ項目抽出部 5 は、定義・参照情報に基づいて、抽出された階層構造をもつデータ項目及びそのデータ項目を構成するデータ項目（子項目）の中から、未使用データ項目を抽出する（不図示）。

【 0 0 3 8 】

集約判定部 6 は、未使用データ項目の抽出結果及びレイアウト結果に基づいて未判定データ項目リストを作成し、未判定データ項目リストを用いて、抽出された未使用データ項目のうちのいくつかをより少数のデータ項目に集約することができるか否か判定する（S 4）。この判定について、詳しくは後述する。

【 0 0 3 9 】

続いて、判定結果通知部 7 は、判定結果を出力するように最適化処理装置 1 が設定されているか否か判定する（S 5）。判定結果を出力するように設定されている場合（S 5 : Y e s）、判定結果通知部 7 は、集約判定部 6 による判定結果をユーザに通知する（S 6）。以下、ユーザに通知される判定結果の内容の一例を挙げる。

【 0 0 4 0 】

- 1) 集約されるデータ項目の項目名、項目長、データ型
- 2) 集約後のデータ項目の項目名、項目長、データ型
- 3) 最適化後のプログラム

判定結果を出力するように設定されていない場合（S 5 : N o）、S 7に進む。

【 0 0 4 1 】

集約可能なデータ項目を集約するように集約指示が最適化処理装置 1 に予め設定されている場合、又は、判定結果通知部 7 が判定結果をユーザに通知した後に、集約指示受付部 8 がユーザから集約すべきデータ項目の指定を受け付けた場合（S 7）、データ項目集約部 9 は、未使用データ項目を未判定データ項目リストに含まれる判定結果に基づいて集約し（S 8）、処理を終了する。

【 0 0 4 2 】

以下、上記 S 1 において作成されるデータ項目辞書及びレイアウトについて、説明する。説明において、図 3 に示す COBOL で記述されたソースプログラムの一例を適宜参照する。なお、以降の説明において用いられたプログラム記述言語は、本発明を適用することができるプログラム記述言語を限定する趣旨ではない。

【 0 0 4 3 】

まず、図 4 から図 7 を用いてデータ項目辞書について説明する。図 4 から図 6 は、図 3 に示すソースプログラムにおいて宣言されたデータ項目に関するデータ項目辞書である。データ項目辞書は、図 4 から図 6 に示すように、データ項目辞書は、ポインタを用いて、ソースプログラムにおいて宣言されたデータ項目間の関係を示す。

【 0 0 4 4 】

図 7 は、データ項目辞書を表記するために図 4 から図 6 において採用された表記方法を説明する図である。図 7（a）に示すように、図 4 から図 6 に示すデータ項目辞書において、各データ項目には 5 つのポインタが与えられている。図 7 中の ϕ は、該当する値が無いことを示す。ポインタの内訳は以下のとおりである。

【 0 0 4 5 】

1）メモリ上で、そのデータ項目の親項目が記録されている位置（アドレス）を示すポインタ

2）メモリ上でのアドレス割付において、そのデータ項目の前に位置するデータ項目が記録されている位置を示すポインタ

3) そのデータ項目の子項目となっているデータ項目の内、メモリ上で先頭となっているデータ項目が記録されている位置を示すポインタ

4) メモリ上でのアドレス割付において、そのデータ項目の次に位置するデータ項目が記録されている位置を示すポインタ

5) 親項目を持たない他のデータ項目が記録されている位置を示すポインタ

なお、子項目とは、階層構造をもつデータ項目を構成するデータ項目をいい、親項目とは、1以上の他のデータ項目によって構成される階層構造をもつそのデータ項目をいう。例えば、図4から図6に示すデータ項目辞書において、データ項目IREC01を構成するデータ項目IDATA11、IDATA12、IDATA13及びIDATA14は、データ項目IREC01の子項目である。逆に、データ項目IREC01は、データ項目IDATA11、IDATA12、IDATA13及びIDATA14の親項目である。

【0046】

また、5)の親項目を持たない他のデータ項目が記録されている位置を示すポインタは、全てのデータ項目について最適化処理を行うために必要である。しかし、このポインタを用いる方法以外の他の方法を採用することとしてもよい。例えば、図7(b)に示すように、ポインタを用いて、データ項目間の階層関係に関係なく全てのデータ項目を直列に連結する事としても良い。

【0047】

次に、図8から図10を用いて、レイアウトについて説明する。まず、図8に、図3に示すCOBOLで記述されたソースプログラムの130行目から170行目までに記述されたコードによって宣言されたデータ項目のレイアウトを示す。図8に示すレイアウトにおいて、データ項目間の間隔の補正は行われていない。

【0048】

図9に、以下に示すCOBOLのコードによって宣言されるデータ項目のレイアウトを示す。以下のコードに示すように、レコードDATA11は、データ項目DATA21及びデータ項目DATA22によって構成される。さらに、データ項目DATA21は、データ項目DATA31、DATA32及びDATA33によって構成され、データ項目DATA22は、データ項目DATA34によって構成される。図9に示すレイアウトにおいて、レイアウト部4によって、遊びビット又は遊びバイトがデータ項目DATA31とDATA32の間及びDA

TA34の後に挿入されている。

【 0 0 4 9 】

```

01      DATA11.
02      DATA21.
03      DATA31 PIC  X(5).
03      DATA32 PICS9(4) BINARY SYNCHRONIZED.
03      DATA33 PICS9(4) BINARY SYNCHRONIZED.
02      DATA22.
03      DATA34 PIC  1(4) BIT    SYNCHRONIZED.

```

図 1 0 に、以下の C のコードによって宣言されるデータ項目のレイアウトを示す。以下のコードに示すように、レコード irec01 は、データ項目 idata11 及び idata12 によって構成される。図 1 0 に示すレイアウトにおいて、文字を格納する変数 idata11 と、整数を格納する変数 idata12 との間には、アラインメントのためにパディングといわれる空のバイトが備えられている。プログラム環境の中には、データ型に応じて、データ項目に割り当てられるメモリ上の先頭アドレスをある整数に割り切れるように要請するものがあり、これをアラインメントという。アラインメントの方式は、処理系に依存する。

【 0 0 5 0 】

```

struct {
    char  idata11[3];
    int   idata12 ;
} irec01;

```

次に、図 2 における S 4 における集約判定処理の概略について説明する。上述の集約判定処理は、以下のように行われる。

【 0 0 5 1 】

1) まず、集約判定部 6 は、階層構造をもつデータ項目、つまりレコードをソースプログラムから抽出する。図 3 に示すソースプログラムの場合、以下のデータ項目が抽出される。

【 0 0 5 2 】

－IREC01、WREC01、WREC02

2) 続いて、集約判定部 6 は、上記 1) で抽出されたデータ項目に対して、データ項目の定義・参照情報に基づいて未使用データ項目を抽出する。図 3 に示すソースプログラムの場合、以下のデータ項目が抽出される。

【 0 0 5 3 】

－IREC01、IDATA11、IDATA12、IDATA13、IDATA14

－WDATA12、WDATA13

－WREC02、WDATA21、WDATA22

3) 上記 2) で抽出された未使用データ項目に対して、データ項目間の連続性を判定する。

【 0 0 5 4 】

3) の判定は、まだ判定されていない未使用データ項目の中（以下、未判定データ項目）から 2 つのデータ項目（以下、項目 A 及び項目 B とする）を選択することによって行われる。ここで、データ項目の集約方法には以下の 2 種類がある。1 つが前後方向の集約であり、もう 1 つが親子方向の集約である。集約方法に対応して判定条件も 2 種類存在する。以下、2 種類の集約について説明する。

【 0 0 5 5 】

a) 前後方向の集約

前後方向の集約とは、データ項目の階層構造において階層レベルが等しい複数のデータ項目を集約することをいう。集約するための条件は以下の通りである。

【 0 0 5 6 】

－親項目が同じである。

－子項目を持たない。

－「項目 A のレイアウト上の相対変位＋項目 A の項目長」と「項目 B のレイアウト上の相対変位」が一致する。一致する場合、項目 A 及び B は、レイアウト中

で隣接する（連続する）。なお、遊びバイト又は遊びビット等の補正が行われている場合、「『項目 A のレイアウト上の相対変位 + 項目 A の項目長 + 補正值』と『項目 B のレイアウト上の相対変位』が一致する」ことが条件となる。

【0057】

前後方向の集約において、集約後のデータ項目は以下ようになる。

－集約後のデータ項目は、無名（データ項目名なし）、又はデータ項目集約部 9 によって生成されたデータ項目名を持つ。

【0058】

－集約後のデータ項目の項目長が集約前のデータ項目の項目長と変わらないように、集約対象となる複数のデータ項目の項目長の合計した値を、集約後のデータ項目のデータ項目長とする。なお、プログラム言語に応じて、遊びバイト又は遊びビット、或いはパディングのように、最適化処理装置 1 におけるレイアウトの補正が行われている場合、この補正も考慮する。

【0059】

－集約後のデータ項目のデータ型は、集約前と同じとするか、又は最小の記憶域サイズを持つデータ型とする。後者の場合、集約対象となる複数のデータ項目のデータ型が互いに同じでない場合にも、集約処理が可能となる。

【0060】

前後方向の集約について図 11 を用いて説明する。

以下のコードによって宣言されるデータ項目のレイアウト及びデータ項目辞書は、図 11 の（a）及び（b）に示すようになる。

【0061】

```
01  A1.
    02  B1      PIC X(10).
    02  B2      PIC X(10).
```

データ項目 B1 と B2 のいずれも未使用データ項目である場合、データ項目 B1 及び B2 について前後方向の集約を行うことにより、上記のコードは以下ようになる

。また、図 11 (a) 及び (b) に示すレイアウト及びデータ項目辞書は、それぞれ図 11 (c) 及び (d) に示すようになる。図 11 (c) 及び (d) から、前後方向の集約によって、データ項目の数が低減されたことがわかる。

【0062】

01 A1.

02 FILLER PIC X(20).

b) 親子方向の集約

親子方向の集約とは、階層構造が異なる（包含関係にある）複数のデータ項目を集約することである。

【0063】

集約するための条件は以下の通りである。

- －項目 A が、項目 B の親項目である。
- －「項目 A の大きさ」と、「項目 B の大きさ」が一致する。

【0064】

親子方向の集約において、集約後のデータ項目は以下のようになる。

- －集約後のデータ項目は、無名（データ項目名なし）、又はデータ項目集約部 9 によって生成されたデータ項目名を持つ。

【0065】

- －集約後のデータ項目の項目長は、項目 B の項目長と同じである。
- －集約後のデータ項目のデータ型は、項目 B のデータ型とする。

親子方向の集約について図 12 を用いて説明する。

【0066】

以下のコードによって宣言されるデータ項目のレイアウト及びデータ項目辞書は、図 12 の (a) 及び (b) に示すようになる。

01 A1.

02 FILLER PIC X(20).

データ項目A1とFILLERのいずれも未使用データ項目である場合、データ項目A1及びFILLERについて親子方向の集約を行うことにより、上記のコードは以下のようになる。また、図12(a)及び(b)に示すレイアウト及びデータ項目辞書は、それぞれ図12(c)及び(d)に示すようになる。図12(c)及び(d)から、親子方向の集約によって、データ項目の数が低減されたことがわかる。

【0067】

01 FILLER PIC X(20).

なお、未使用データ項目が2層以上の階層構造を持つ場合がある。この場合、1回の最適化処理では、処理が不十分になってしまう可能性がある。従って、上記の判定処理は繰り返して行なわれる必要がある。繰り返して処理を行う場合、抽出された未使用データ項目すべてに対して、以下の考え方に基づいて最適化処理を行う。

【0068】

一階層の深いデータ項目を優先させて集約させる。

一親子方向の集約判定を行ってから前後方向の集約判定を行うという順序で再帰的に判定する。

【0069】

以下、実際にコードを参照しながら、判定処理を繰り返すことにより2層以上の階層構造をもつデータ項目がどのように処理されるのかについて説明する。

まず、以下のような初期コードを仮定する。説明のために、このコードにおいて宣言されるすべてのデータ項目は未使用データ項目であると仮定する。

【0070】

01 A1.

02 B1 PIC X(20).

02 B2.

```
03 C1      PIC X(10).  
03 C2      PIC X(10).  
02 B3      PIC X(20).
```

1 回目の最適化処理において、最も階層が深いデータ項目C1及びC2が集約される。従って、上記の初期コードは以下のようになる。

【 0 0 7 1 】

```
01 A1.  
02 B1      PIC X(20).  
02 B2.  
03 FILLER  PIC X(20).  *> 1 回目:前後方向の集約  
02 B3      PIC X(20).
```

さらに、1 回目の最適化処理後のコードについて、2 回目の最適化処理を行うと、データ項目B2及びFILLER項目が集約される。その結果、以下に示す2 回目の最適化処理後のコードが得られる。

【 0 0 7 2 】

```
01 A1.  
02 B1      PIC X(20).  
02 FILLER  PIC X(20).  *> 2 回目:親子方向の集約  
02 B3      PIC X(20).
```

さらに、2 回目の最適化処理後のコードについて、3 回目の最適化処理を行うと、データ項目B1及びFILLER項目が集約される。その結果、以下に示す3 回目の最適化処理後のコードが得られる。

【 0 0 7 3 】

01 A1.

02 FILLER PIC X(40). *>3 回目:前後方向の集約

02 B3 PIC X(20).

さらに、3 回目の最適化処理後のコードについて、4 回目の最適化処理を行うと、FILLER項目及びデータ項目B3が集約される。その結果、以下に示す4 回目の最適化処理後のコードが得られる。

【0 0 7 4】

01 A1.

02 FILLER PIC X(60). *>4 回目:前後方向の集約

最後に、4 回目の最適化処理後のコードについて、5 回目の最適化処理を行うと、データ項目A1及びFILLER項目が集約される。その結果、以下に示す5 回目の最適化処理後のコードが得られる。

【0 0 7 5】

01 FILLER PIC X(60). *>5 回目:親子方向の集約

5 回目の最適化処理後のコードをこれ以上集約することはできないため、処理は終了する。

【0 0 7 6】

以下、集約判定部6による集約判定についてより詳しく説明する。集約判定は、図2のS4の直前において未使用データ項目抽出部5によって抽出された未使用データ項目について行われる。集約判定において、集約判定部6は、未判定データ項目及び判定結果を管理するための未判定データ項目リストを利用する。未判定データ項目リストは、未判定データ項目及び判定結果を示す情報と、データ項目辞書へのポインタと、次に判定対象とするべきデータ項目へのポインタを含む。

【0077】

図13及び14に、図3に示すソースプログラムに係わる未判定データ項目リストのポインタを示す。図13及び14において、 ϕ は該当する値がないことを示す。図15に、図13および14において用いられたポインタの表記法及び、未使用データ項目リストに含まれるポインタ以外の情報の内訳を示す。図15に示すように、未使用データ項目リストに含まれる情報として、以下が挙げられる。

【0078】

- 1) 未判定であるか否かを示す情報（以下、状態情報という）
- 2) 判定の結果、集約可能であるのか、削除可能であるのかを示す情報（以下、判定結果情報という）
- 3) 集約後のデータ項目の項目長（以下、項目長情報という）
- 4) 集約後のデータ項目のデータ型（以下、型情報という）
- 5) 項目A及び項目Bとして設定されたデータ項目の項目名

図17及び18において、判定対象とするべきデータ項目としてポイントされているデータ項目は、図2に示すS4の直前において、未使用データ項目抽出部5によって未使用データ項目として抽出されたデータ項目である。集約判定の対象となるデータ項目は、未判定データ項目リストに含まれるポインタに基づいて選定される。また、未判定データ項目リストに含まれる判定結果は、データ項目を集約する際にデータ項目集約部9によって参照される。

【0079】

以下、図16から図19に示すフローチャートを用いて、集約判定において行われる処理の流れについて説明する。

集約判定において、まず、集約判定部6は、未判定データ項目リスト内のデータ項目についての状態情報を「未判定」とすることにより、判定処理の対象となる項目A及びBが設定されていない状態にする（S11及びS12）。続いて、集約判定部6は、未判定データ項目リスト内の状態情報を参照する事により、未判定データ項目があるか否か判定する（S13）。全てのデータ項目について集約判定を行った場合（S13：No）、図19のS42に進む（後述）。未判定

データ項目がある場合、集約判定部 6 は、未判定データ項目リストに含まれるポインタに基づいて、未判定データ項目の中から 1 つのデータ項目を取り出す（S 1 4）。例えば、図 3 に示すソースプログラムを最適化する場合、最初は、データ項目 IREC01 が取出されることになる。

【 0 0 8 0 】

続いて、集約判定部 6 は、未判定データ項目リスト内の状態情報に基づいて項目 A が既に設定されているか否か判定する（S 1 5）。まだ項目 A が設定されていない場合（S 1 5 : Y e s）、集約判定部 6 は、未判定データ項目リスト内の取り出されたデータ項目についての状態情報を「判定済み」に設定することにより、そのデータ項目を項目 A として設定する（S 1 6）。S 1 6 の設定後、処理は S 1 3 に戻る。項目 A が既に設定されている場合、集約判定部 6 は、取出されたデータ項目を項目 B として設定する（S 1 7）。S 1 7 の設定は、S 1 6 の設定と同様であるため説明を省略する。

【 0 0 8 1 】

項目 A 及び項目 B を設定した後、集約判定部 6 は、データ項目辞書に基づいて、項目 A は項目 B の親項目であるか否か判定する（S 1 8）。項目 A が項目 B の親項目である場合（S 1 8 : Y e s）、集約判定部 6 は、未判定データ項目リスト内の項目 B として設定されているデータ項目についての状態情報を「未判定」に戻すことにより、項目 B を未設定の状態に戻す。さらに、集約判定部 6 は、未判定データ項目リスト内の、次の判定対象とすべき項目へのポインタを引数として再帰呼出しを行う。そして、再帰呼出しから復帰後、図 1 7 に示す S 2 0 に進む。項目 A が項目 B の親項目でない場合（S 1 8 : N o）、集約判定部 6 は、図 1 8 に示す S 2 7 に進む。

【 0 0 8 2 】

S 2 0 において、再帰呼出しから復帰後、集約判定部 6 は、未判定データ項目リスト及びデータ項目辞書に基づいて、項目 A が項目 B 以外の子項目を持たないか否か判定する。ここで、集約判定部 6 は、データ項目辞書に含まれる子項目であっても、未判定データ項目リスト内の判定結果情報が「削除可」に設定されている子項目を、存在しない子項目として扱う。

【0083】

項目Aが項目B以外の子項目を持たない場合（S20：Yes）、S21に進み、項目Aと項目Bについて親子方向の集約を行う。

S21において、集約判定部6は、未判定データ項目リスト内の項目Aの型情報に、項目Bのデータ型を設定する。さらに、集約判定部6は、未判定データ項目リスト内の項目Aの判定結果情報に「集約可」を設定し（S22）、項目Bの判定結果情報に「削除可」を設定する（S23）。また、集約判定部6は、項目Aの項目長情報に項目Bの項目長を設定する。続いて、集約判定部6は、項目Bを未設定として（S24）、図16のS13に戻る。

【0084】

項目Aが項目B以外の子項目を持つ場合（S20：No）、集約判定部6は、項目A及び項目Bを未設定として（S25及びS26）、図16のS13に戻る。

【0085】

上述のS18において、項目Aが項目Bの親項目でない場合（S18：No）、S27において、集約判定部6は、項目Aと項目Bの親項目が同じであるか否か判定する。項目Aと項目Bの親項目が同じである場合（S27：Yes）、S28に進み、そうでない場合（S27：No）、図19のS38に進む。

【0086】

S28において、集約判定部6は、未判定データ項目リスト及びデータ項目辞書に基づいて、項目Bは子項目を持つか否か判定する（S28）。なお、S28の判定は、S20の判定と同様である。

【0087】

項目Bが子項目を持つ場合（S28：Yes）、未判定データ項目リスト内の、次の判定対象とすべきデータ項目へのポインタを引数として再帰呼出しを行った後（S29）、S30に進む。項目Bが子項目を持たない場合（S28：No）、S29を行わないでS30に進む。項目Bが子項目を持つ場合（S30：No）、S37に進む。

【0088】

S 3 0において、集約判定部 6 は、項目 B が子項目を持たないか否か判定する。項目 B が子項目を持たない場合（S 3 0 : Y e s）、集約判定部 6 は、さらに、レイアウト部 4 によって作成されたレイアウトに基づいて、項目 A を格納する領域と項目 B を格納する領域とが、連続する領域であるか否か判定する（S 3 1）。項目 A を格納する領域と項目 B を格納する領域とが、連続する領域である場合、S 3 2 に進み、項目 A 及び項目 B について前後方向の集約を行う。そうでない場合、S 3 7 に進む。

【0089】

S 3 2において、集約判定部 6 は、レイアウトに示されるデータ項目のデータ項目長に基づいて、項目 A の項目長に項目 B の項目長を加算することにより、集約後のデータ項目の項目長を算出する。そして、集約判定部 6 は、項目 A の項目長情報に、算出した項目長を設定する。なお、レイアウト時に遊びビット又は遊びバイト、パディング等の補正がされている場合、項目 A の項目長と項目 B の項目長の和に、その補正値を更に加算して集約後の項目長を算出する。

【0090】

続いて、集約判定部 6 は、未判定データ項目リスト内の項目 A の型情報を最小の記憶域サイズを持つデータ型に設定する（S 3 3）。これにより、集約後のデータ項目のデータ型が決定される。なお、項目 A と項目 B のデータ型が同じである場合、データ型を変更しないこととしても良い。

【0091】

さらに、集約判定部 6 は、未判定データ項目リスト内の項目 A の判定結果情報に「集約可」を設定し（S 3 4）、項目 B の判定結果情報に「削除可」を設定する（S 3 5）。その後、集約判定部 6 は、項目 B を未設定として（S 2 4）、図 1 6 の S 1 3 に戻る。

【0092】

一方、S 3 0 又は S 3 1 の判定で N o であった場合、S 3 7 において、集約判定部 6 は、項目 A として、現在項目 B として設定されているデータ項目を設定しなおし、項目 B を未設定とする。その後、図 1 6 の S 1 3 に戻る。

【0093】

上述の S 2 7 の判定で N o である場合、集約判定部 6 は、再帰呼出しをされていないか否か判定する (S 3 8)。再帰呼出しをされていない場合 (S 3 8 : Y e s)、集約判定部 6 は、項目 B として設定されているデータ項目を項目 A として設定しなおし (S 3 9)、さらに、項目 B を未設定として設定する (S 4 0)。その後、図 1 6 の S 1 3 に戻る。再帰呼出しをしている場合 (S 3 8 : N o)、集約判定部 6 は、項目 B に設定されているデータ項目の状態情報を「未判定」に設定し、処理は再帰呼出しの呼び出し元に戻る (S 4 1)。さらに、処理は呼び出し元、つまり再帰呼出しの呼び出し元又はメインフローに戻り (S 4 2)、判定処理を終了する。

【0094】

図 2 0 から 2 7 に、図 3 に示すソースプログラムについて集約判定処理を行う場合の各段階における、項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す。図 2 0 から 2 7 において、段階を識別する段階番号、判定処理中の手順を識別するステップ番号、そのステップを行う際に集約判定部 6 によって取出されて一次格納域に格納されているデータ項目の項目名、項目 A として設定されているデータ項目の項目名、項目 B として設定されているデータ項目の項目名、未判定データ項目リスト内の状態情報及び判定結果情報を示す。図 2 0 から 2 7 において、φ は「該当無し」を示す。また、ステップ番号の後に記された括弧内の数字は、再帰しているのか否かを示す。括弧内の数字が「0」である場合は、再帰していないことを示し、「1」である場合は、再帰している（後で呼び出し元のステップに戻る必要がある）ことを示す。

【0095】

図 2 0 から 2 7 における未判定データ項目リストの表記法は以下のとおりである。状態情報及び判定結果情報以外の情報については、記載を省略する。

データ項目名 (状態情報, 判定結果情報)

－状態情報： 未：未判定

済：判定済

－判定結果情報： 集：集約可

削：削除可

例えば、IREC01（済，集）と表記されている場合、データ項目IREC01は、「判定済み」であり、判定結果は「集約可」であることを示す。

【 0 0 9 6 】

以下、図 2 0 から 2 7 を参照して具体的に集約判定について説明する。

例えば、段階 1 の S 0 0 において（判定処理を行う前）、未判定データ項目の内訳は以下のとおりである。

【 0 0 9 7 】

IREC01、IDATA11、IDATA12、IDATA13、IDATA14、WDATA12、WDATA13、WREC02、WDATA21、WDATA22

続いて、段階 2 及び段階 3 において、それぞれステップ S 1 1 及び S 1 2 を実行することにより項目 A 及び項目 B が未設定となり、段階 4 において S 1 4 を実行することによりデータ項目IREC01が取出され、メモリ上の一時格納域に格納される。段階 5 において S 1 5 を実行することにより、IREC01の状態情報は「未」から「済」に設定される。その後、さらに段階 6 から 7 において、未判定データ項目リストからデータ項目IDATA11が取出されて項目 B として設定された後、再帰呼出しが行われ、処理は S 0 （再帰中）に移行する。

【 0 0 9 8 】

段階 8 から 1 4 にかけて、項目 A 及び項目 B が設定される。図 2 1 の段階 1 4 において示されるように、項目 A 及び項目 B として設定されたデータ項目はそれぞれIDATA11及びIDATA12である。図 1 3 に示すデータ項目辞書から明らかなように、IDATA11及びIDATA12は親子関係ではなく、且つ、IDATA12は子項目を持たないため、段階 1 5 から 1 7 にかけて S 2 7 から S 3 6 が行われる。その結果、図 2 2 に示すように、段階 1 7 では、IDATA11の状態情報及び判定結果情報はそれぞれ「済」及び「集」として設定され、IDATA12の状態情報及び判定結果情報はそれぞれ「済」及び「削」として設定される。つまり、IDATA11とIDATA12について前後方向の集約が可能であると設定される。さらに、項目 B は未設定の状態にされる。

【 0 0 9 9 】

図 2 2 の段階 1 7 の後、図 2 2 の段階 1 8 から段階 2 1 にかけて、上記と同様

にして S 1 3 から S 3 6 が実行されることにより、IDATA11とIDATA13について前後方向の集約が可能であると設定される。さらに、図 2 3 の段階 2 2 から 2 5 にかけて、上記と同様にして IDATA11とIDATA14について前後方向の集約が可能であると設定される。図 2 3 の段階 2 5 の後、図 2 3 の段階 2 6 及び 2 7 にかけて S 1 3 から S 1 7 が実行されることにより、WDATA12が項目 B として設定される。IDATA11はWDATA12の親項目ではなく（S 1 8 の判定で N o）、かつ、IDATA11とWDATA12の親項目は同じではないため（S 2 7 の判定で N o）、図 2 3 の段階 2 8 において S 4 1 を実行することにより、呼出し元である S 1 9 に復帰する。その時の項目 A 及び項目 B は、再帰呼出し時に設定された IREC01及びIDATA11である。

【0100】

IREC01はIDATA11の親項目であり、且つ、上記において前後関係の集約を行った結果、IDATA11以外の子項目を持たない。従って、図 2 4 の段階 2 9 から段階 3 1 において、S 2 2 から S 2 4 が実行される。その結果、図 2 4 に示すように、段階 3 1 では、IREC01の状態情報及び判定結果情報はそれぞれ「済」及び「集」として設定され、IDATA11の状態情報及び判定結果情報はそれぞれ「済」及び「削」として設定される。つまり、IREC01とIDATA11について親子方向の集約が可能であると判定される。さらに、項目 B は未設定の状態にされる。ここまでの集約判定によって、データ項目 IREC01、IDATA11、IDATA12、IDATA13及びIDATA14 は、1 つのデータ項目に集約可能であると判定される。

【0101】

図 2 4 の段階 3 1 の後、図 2 4 の段階 3 2 及び段落 3 3 において S 1 3 から S 1 7 を実行することにより、項目 A として IREC01を設定し、項目 B として WDATA12を設定する。しかし、IREC01はWDATA12の親項目ではなく、且つ、IREC01とWDATA12の親項目は同じではないため、これらの項目を集約する事はできないと判定される。その結果、図 2 4 の段階 3 4 及び段階 3 5 において、S 3 9 及び S 4 0 を実行することにより、WDATA12を項目 A として設定しなおし、さらに項目 B を未設定として、S 1 3 に戻る。これ以降、同様にして、図 2 5 の段階 3 6 から図 2 6 の段階 4 0 にかけて、データ項目 WREC01を構成する 4 つのデータ項目のうち

WDATA12とWDATA13について前後方向の集約を行う事が可能であると判定される。さらに、図 2 6 の段階 4 1 から図 2 7 の段階 5 9 かけて、データ項目 WREC02と、そのデータ項目を構成する 2 つのデータ項目について前後方向及び親子方向の集約を行う事が可能であると判定される。

【 0 1 0 2 】

未判定データ項目リストに格納された判定結果は、判定結果通知部 7 によってユーザに通知される。集約指示受付部 8 がユーザから集約指示を受け付けた場合、データ項目集約部 9 によって、判定結果に基づいてデータ項目を集約することによりソースプログラムの最適化が行われる。

【 0 1 0 3 】

図 2 8 に、図 3 に示すソースプログラムを最適化した結果得られたプログラムを示す。図 3 に示すソースプログラムの 1 3 0 行目から 1 7 0 行目において以下の 5 つのデータ項目が宣言されている。図 3 に示すソースプログラムの処理部の記述から明らかなように、これらのデータ項目は未使用データ項目である。

【 0 1 0 4 】

- IREC01
- DATA11 (データ型：文字データ、項目長：20)
- DATA12 (データ型：文字データ、項目長：20)
- DATA13 (データ型：文字データ、項目長：20)
- DATA14 (データ型：文字データ、項目長：20)

図 2 8 の 1 3 0 行目に示すように、集約の結果、これらのデータ項目は、データ型が文字データであり、項目長が 8 0 である 1 つの FILLER 項目に置き換えられている。つまり、5 つのデータ項目が 1 つのデータ項目になったため、項目数が削減されていることが分かる。

【 0 1 0 5 】

同様に、図 3 に示すソースプログラムの 2 1 0 行目及び 2 2 0 行目において宣言された、データ型が数値データであり、それぞれ項目長が 2 である 2 つのデータ項目 WDATA12 及び WDATA13 は、最適化処理の結果、データ型が数値データであり、項目長が 4 である 1 つの FILLER 項目に集約されている。また、同様に、図 3 に

示すソースプログラムの240行目から260行目において宣言された3つのデータ項目は、1つのFILLER項目に集約されている。

【0106】

なお、この例では、集約される複数のデータ項目のデータ型が同じである場合、集約後のデータ項目のデータ型も集約前と同じデータ型となるとしているが、集約前のデータ項目のデータ型に関係なく、集約後のデータ項目のデータ型を記憶域サイズが最も小さなデータ型とすることとしてもよい。いずれの場合であっても、集約後のFILLER項目の項目長は、集約されたデータ項目の項目長の合計に等しい長さに設定されるため、集約によってデータ項目のメモリ上のアドレスが変化することは無い。従って、レコードの一部となっている複数のデータ項目を1つのデータ項目に集約しても、データ項目のメモリ上のアドレスがずれるという問題は生じない。

【0107】

上記の最適化処理装置1は、未使用データ項目を定義するコードを削除する機能を実現する未使用データ項目削除部（不図示）を更に備えることとしてもよい。この未使用データ項目定義するコードを削除する機能は、従来からコンパイラ等に与えられているため、詳しい説明を省略する。

【0108】

この未使用データ項目削除部を備える最適化処理装置1において、最適化処理を行う対象となるソースプログラムに対して、データ項目の集約判定を行う前に、未使用データ項目削除部は、未使用データ項目を宣言するコードをソースプログラムから削除することにより、集約判定のための前処理を行い、前処理が行われたソースプログラムをプログラム格納部11に格納する。この前処理を行うことにより、集約判定の対象となる未使用データ項目の数を低減させておく。

【0109】

続いて、上記と同様にして、データ項目抽出部2、データ項目辞書作成部3、レイアウト部4、未使用データ項目抽出部5、集約判定部6及びデータ項目集約部9は、プログラム格納部11から取出された前処理のすんだソースプログラム中の未使用データ項目を集約する。さらに、未使用データ項目削除部によって、

その未使用データ項目が集約されたソースプログラムから、再度、未使用データ項目を宣言するコードをソースプログラムから削除する。

【 0 1 1 0 】

図 2 9 に、図 3 に示すソースプログラムに未使用データ項目の集約を行った後に不要な定義コードを削除した結果を示す。図 2 9 から、図 3 4 に示す従来の技術による不要な定義コードの削除結果と比べて、最適化処理装置 1 の方が、ソースプログラム中に残る未使用データ項目の数が少ないことが分かる。

【 0 1 1 1 】

図 3 0 に、C 言語で記述されたソースプログラムの一例を示す。図 3 0 に示すソースプログラムは、図 3 に示す COBOL で記述されたソースプログラムに対応する内容をもつ。COBOL で記述されたソースプログラムと同様に、最適化処理装置 1 によって C 言語で記述されたソースプログラム中の未使用データ項目を集約することができる。未使用データ項目を集約することによって図 3 0 に示すソースプログラムを最適化すると、図 3 1 に示すようなソースプログラムが得られる。なお、C 言語には、COBOL の FILLER 項目に相当するデータ項目が無いため、図 3 0 及び図 3 1 において、集約後のデータ項目の項目名は、集約されるデータ項目のなかで、レイアウト上で最先のアドレスを持つデータ項目の項目名と同じすることとした。また、データ型 short（短精度整数データ）は、2 バイトであると仮定している。

【 0 1 1 2 】

図 3 0 及び図 3 1 に示すように、C 言語で記述されたソースプログラムであっても、COBOL 言語で記述されたソースプログラムと同様に未使用データ項目の数を低減することができる。つまり、最適化処理装置 1 は、言語に依存しないで最適化処理を行う事が可能である。

【 0 1 1 3 】

上記において説明した最適化処理装置 1 は、コンピュータを用いて構成することができる。以下、コンピュータの構成について、図 3 2 を用いて説明する。

図 3 2 に示すようにコンピュータ 2 0 は、CPU 2 1、メモリ 2 2、入力装置 2 3、出力装置 2 4、外部記憶装置 2 5、媒体駆動装置 2 6、及びネットワーク

接続装置 2 7 を備え、それらはバス 2 8 により互いに接続されている。

【 0 1 1 4 】

メモリ 2 2 は、例えば、ROM (Read Only Memory)、RAM (Random Access Memory) 等を含み、処理に用いられるプログラムとデータを格納する。各格納部 1 5、1 6 及び 1 7 から読み出されたソースプログラムやデータは、一時的にメモリ 2 2 に格納される。また、データ項目抽出部 2 によって抽出されたデータ項目は、レイアウト部 4 によってこのメモリ 2 2 にレイアウトされる。最適化処理装置 1 を構成するデータ項目抽出部 2、データ項目辞書作成部 3、レイアウト部 4、未使用データ項目抽出部 5、集約判定部 6、判定結果通知部 7、集約指示受付部 8 及びデータ項目集約部 9 は、コンピュータ 2 0 のメモリ 2 2 の特定のプログラムコードセグメントにプログラムとして格納される。なお、場合によっては、さらに未使用データ項目削除部が、コンピュータ 2 0 のメモリ 2 3 のプログラムコードセグメントにプログラムとして格納されることとしてもよい。CPU 2 1 は、メモリ 2 2 を利用して上述のプログラムを実行することにより、必要な処理を行う。

【 0 1 1 5 】

入力装置 2 3 は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、ユーザからの指示や情報の入力に用いられる。出力装置 2 4 は、例えば、ディスプレイやプリンタ等であり、コンピュータのユーザへの問い合わせ、処理結果等の出力に用いられる。入力装置 2 3 及び出力装置 2 4 は、図 1 の入出力装置 1 0 に対応する。

【 0 1 1 6 】

外部記憶装置 2 5 は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク装置等である。外部記憶装置 2 5 は、プログラム格納部 1 1、データ項目辞書格納部 1 2 及び未判定データ項目リスト格納部 1 3 を実現する。さらに、上述のプログラムをコンピュータ 2 0 の外部記憶装置 2 5 に保存しておき、必要に応じて、それらをメモリ 2 2 にロードして使用することもできる。

【 0 1 1 7 】

媒体駆動装置 2 6 は、可搬記録媒体 2 9 を駆動し、その記録内容にアクセスす

る。可搬記録媒体 2 9 としては、メモリカード、メモリスティック、フレキシブルディスク、C D - R O M (Compact Disc Read Only Memory)、光ディスク、光磁気ディスク、D V D (Digital Versatile Disk) 等、任意のコンピュータ読み取り可能な記録媒体が用いられる。この可搬記録媒体 2 9 に上述のプログラムを格納しておき、必要に応じて、それをコンピュータ 2 0 のメモリ 2 2 にロードして使用することもできる。

【 0 1 1 8 】

ネットワーク接続装置 2 7 は、L A N、W A N 等の任意のネットワーク（回線）を介して外部の装置と通信し、通信に伴うデータ変換を行う。また、必要に応じて、上述のプログラムを外部の装置から受け取り、それをコンピュータ 2 0 のメモリ 2 2 にロードして使用することもできる。

【 0 1 1 9 】

図 3 3 は、図 3 2 のコンピュータへのプログラムのローディングを説明している。

上記最適化処理装置 1 に相当する機能をコンピュータに実現させるためのプログラムを、直接コンピュータの入力装置 2 3 から入力することとしてもよいが、以下のようにしてコンピュータにロードする事もできる。例えば、コンピュータで読み取り可能な記録媒体 2 9 に上記プログラムを予め記憶させておく。そして、図 3 3 に示すようにしてその記録媒体 2 9 からプログラムをコンピュータに読み出させて、そのコンピュータのメモリ 2 2 や外部記憶装置 2 5 に一旦格納させる。コンピュータに最適化処理を行わせるためには、この格納されたプログラムをそのコンピュータの有する C P U 2 1 に読み出させて、そのコンピュータに実行させるように構成すればよい。

【 0 1 2 0 】

また、プログラム（データ）提供者 3 0 が有する D B から、通信回線（ネットワーク） 3 1 を介して、プログラムをコンピュータにダウンロードすることとしてもよい。この場合、例えば、プログラム（データ）提供者 3 0 が有する送信側コンピュータでは、プログラムを表現するプログラム・データをプログラム・データ・シグナルに変換し、モデムを用いて変換されたプログラム・データ・シグ

ナルを変調することにより伝送信号を得て、得られた伝送信号を通信回線 3 1 (伝送媒体) に出力する。プログラムを受信するコンピュータでは、モデムを用いて受信した伝送信号を復調することにより、プログラム・データ・シグナルを得て、得られたプログラム・データ・シグナルを変換することにより、プログラム・データを得る。なお、送信側のコンピュータと受信側のコンピュータの間を接続する通信回線 3 1 (伝送媒体) がデジタル回線の場合、プログラム・データ・シグナルを通信することも可能である。

【 0 1 2 1 】

以上、本発明の実施形態について説明したが、本発明は上述した実施形態及び変形例に限定されるものではなく、その他の様々な変更が可能である。例えば、最適化処理装置 1 をコンパイラ等に適用することとしてもよい。これにより、コンパイルに必要なメモリ容量を低減させ、効率的にコンパイルを行う事が可能となる。また、本最適化処理装置 1 を、データ項目辞書を作成するために使用することとしてもよい。

【 0 1 2 2 】

(付記 1) プログラムを最適化する制御をコンピュータに行わせるコンピュータ・プログラムであって、

前記プログラムからデータ項目を抽出し、

前記データ項目を前記コンピュータに備えられたメモリ上にレイアウトし、

抽出された前記データ項目から、定義されたが使用されない未使用データ項目を抽出し、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をもつデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新たなデータ項目に集約することが可能か否か判定し、

前記判定結果に基づいて、前記複数のデータ項目を前記1つの新たなデータ項目に集約したプログラムを出力する、

ことを含む制御を前記コンピュータに行わせることを特徴とするコンピュータ・プログラム。

【 0 1 2 3 】

(付記2) 集約される前記複数の未使用データ項目は、前記メモリ上において互いに隣接する領域にレイアウトされている、

ことを特徴とする付記1に記載のコンピュータ・プログラム。

【0124】

(付記3) 集約される前記複数の未使用データ項目は、共に階層構造をもつ他のデータ項目を構成するデータ項目であって、且つ、前記階層構造における階層レベルが互いに同じである、

ことを特徴とする付記2に記載のコンピュータ・プログラム。

【0125】

(付記4) 集約される前記複数の未使用データ項目は、階層構造をもつデータ項目と当該データ項目を構成するデータ項目であり、且つ、前記階層構造をもつデータ項目は1つのデータ項目によって構成される、

ことを特徴とする付記1に記載のコンピュータ・プログラム。

(付記5) 前記集約において、複数の未使用データ項目を宣言するコードを前記プログラムから削除し、1つの新たなデータ項目を宣言するコードを追加する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記1に記載のコンピュータ・プログラム。

【0126】

(付記6) 集約される前記複数の未使用データ項目の項目長に基づいて前記新たなデータ項目の項目長を設定する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記5に記載のコンピュータ・プログラム。

【0127】

(付記7) 集約される前記複数の未使用データ項目のデータ型が全て同じである場合、集約される前記複数の未使用データ項目のデータ型と同じデータ型として、前記新たなデータ項目のデータ型を設定する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記5に記載のコンピュータ・プログラム。

【 0 1 2 8 】

（付記 8） 前記新たなデータ項目のデータ型を、記憶域サイズが最小となるデータ型として設定する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記 5 に記載のコンピュータ・プログラム。

【 0 1 2 9 】

（付記 9） 前記新たなデータ項目の項目名を無名として設定する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記 5 に記載のコンピュータ・プログラム。

【 0 1 3 0 】

（付記 1 0） 前記新たなデータ項目の項目名を、集約される前記複数の未使用データ項目のうちの何れか 1 つに基づいて設定する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記 5 に記載のコンピュータ・プログラム。

【 0 1 3 1 】

（付記 1 1） 変更された前記プログラムに基づいて、前記メモリ上の前記データ項目のレイアウトを変更する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記 1 に記載のコンピュータ・プログラム。

【 0 1 3 2 】

（付記 1 2） 変更された前記プログラムから、未使用データ項目を定義するコードを削除する、

ことを更に含む制御を前記コンピュータに行わせることを特徴とする付記 1 に記載のコンピュータ・プログラム。

【 0 1 3 3 】

（付記 1 3） プログラムを最適化する制御をコンピュータに行わせるプログラムを記録した記録媒体であって、

前記プログラムからデータ項目を抽出し、

前記データ項目を前記コンピュータに備えられたメモリ上にレイアウトし、
抽出された前記データ項目から、定義されたが使用されない未使用データ項目
を抽出し、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をも
つデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新
たなデータ項目に集約することが可能か否か判定し、

前記判定結果に基づいて、前記複数のデータ項目を前記1つの新たなデータ項
目に集約したプログラムを出力する、

ことを含む制御を前記コンピュータに行わせるプログラムを記録した記録媒体

【0134】

(付記14) コンピュータが、プログラムを最適化するプログラム最適化方
法であって、

前記プログラムからデータ項目を抽出し、

前記データ項目を前記コンピュータに備えられたメモリ上にレイアウトし、
抽出された前記データ項目から、定義されたが使用されない未使用データ項目
を抽出し、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をも
つデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新
たなデータ項目に集約することが可能か否か判定し、

前記判定結果に基づいて、前記複数のデータ項目を前記新たなデータ項目に集
約するように前記プログラムを変更する、

ことを含むことを特徴とするプログラム最適化方法。

【0135】

(付記15) プログラムを最適化する最適化処理装置であって、
前記プログラムからデータ項目を抽出するデータ項目抽出手段と、
抽出された前記データ項目をメモリ上にレイアウトするレイアウト手段と、
抽出された前記データ項目から、定義されたが使用されない未使用データ項目
を抽出する未使用データ項目抽出手段と、

前記レイアウト結果に基づいて、前記未使用データ項目のうち、階層構造をもつデータ項目の少なくとも一部を構成する複数の未使用データ項目を、1つの新たなデータ項目に集約することが可能か否か判定する集約判定手段と、

前記判定結果に基づいて、前記複数のデータ項目を前記1つの新たなデータ項目に集約したプログラムを出力するデータ項目集約手段と、

を備えることを特徴とする最適化処理装置。

【0136】

【発明の効果】

以上詳細に説明したように、本発明によれば、階層構造における階層レベル及びメモリ上でのデータ項目の配置に基づいて、階層構造を有するデータ項目を構成する複数の未使用データ項目を1つのデータ項目に集約することにより、未使用データ項目の数を低減することが可能となる。延いては、コンパイル時やデータ項目辞書作成時に必要となるメモリ量を低減するようにプログラムを最適化することが可能となる。

【図面の簡単な説明】

【図1】

最適化処理装置の構成図である。

【図2】

プログラムの最適化処理の大まかな流れを示すフローチャートである。

【図3】

COBOL で記述されたソースプログラムの一例を示す図である。

【図4】

図3に示すソースプログラムのデータ項目辞書を示す図（その1）である。

【図5】

図3に示すソースプログラムのデータ項目辞書を示す図（その2）である。

【図6】

図3に示すソースプログラムのデータ項目辞書を示す図（その3）である。

【図7】

データ項目辞書の表記方法を説明する図である。

【図 8】

データ項目のレイアウトの一例を示す図（その 1）である。

【図 9】

データ項目のレイアウトの一例を示す図（その 2）である。

【図 1 0】

データ項目のレイアウトの一例を示す図（その 3）である。

【図 1 1】

前後方向の集約を説明する図である。

【図 1 2】

親子方向の集約を説明する図である。

【図 1 3】

未判定データ項目リストに含まれるポインタを説明する図（その 1）である。

【図 1 4】

未判定データ項目リストに含まれるポインタを説明する図（その 2）である。

【図 1 5】

図 1 3 及び図 1 4 に示す未判定データ項目リストに含まれるポインタの表記方法及び未判定データ項目リストに含まれる情報を説明する図である。

【図 1 6】

集約判定の手順を示すフローチャート（その 1）である。

【図 1 7】

集約判定の手順を示すフローチャート（その 2）である。

【図 1 8】

集約判定の手順を示すフローチャート（その 3）である。

【図 1 9】

集約判定の手順を示すフローチャート（その 4）である。

【図 2 0】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 1）である。

【図 2 1】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 2）である。

【図 2 2】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 3）である。

【図 2 3】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 4）である。

【図 2 4】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 5）である。

【図 2 5】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 6）である。

【図 2 6】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 7）である。

【図 2 7】

図 3 に示すソースプログラムについて集約判定を行う場合の各段階における項目 A 及び項目 B の設定内容並びに未判定データ項目リストの内容を示す図（その 8）である。

【図 2 8】

図 3 に示すソースプログラムを最適化した結果を示す図である。

【図 2 9】

図 2 8 に示す最適化されたソースプログラムから付与運当て意義コードを削除した結果を示す図である。

【図 3 0】

C で記述されたソースプログラムの一例を示す図である。

【図 3 1】

図 3 0 に示すソースプログラムを最適化した結果を示す図である。

【図 3 2】

コンピュータの構成図である。

【図 3 3】

データやプログラムのコンピュータへのローディングを説明する図である。

【図 3 4】

COBOL で記述されたソースプログラムの一例を示す図である。

【図 3 5】

従来技術を用いて図 3 4 に示すソースプログラムから未使用データ項目を定義するコードを削除した結果を示す図である。

【図 3 6】

C で記述されたソースプログラムの一例を示す図である。

【図 3 7】

従来技術を用いて図 3 9 に示すソースプログラムから未使用データ項目を定義するコードを削除した結果を示す図である。

【符号の説明】

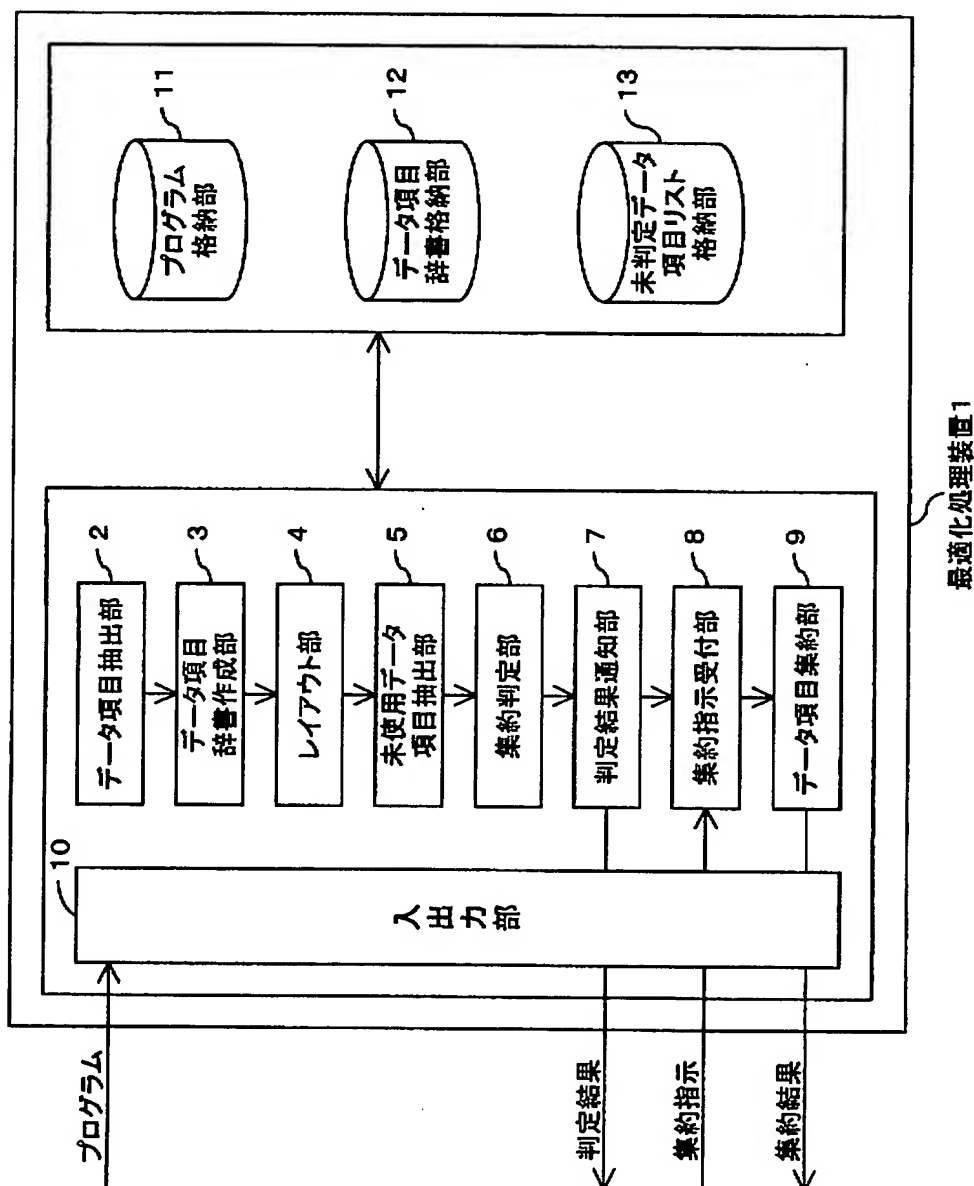
- 1 最適化処理装置
- 2 データ項目抽出部
- 3 データ項目辞書作成部
- 4 レイアウト部
- 5 未使用データ項目抽出部
- 6 集約判定部

- 7 判定結果通知部
- 8 集約指示受付部
- 9 データ項目集約部
- 1 0 入出力部
- 1 1 プログラム格納部
- 1 2 データ項目辞書格納部
- 1 3 未判定データ項目リスト格納部
- 2 0 コンピュータ
- 2 1 C P U
- 2 2 メモリ
- 2 3 入力装置
- 2 4 出力装置
- 2 5 外部記憶装置
- 2 6 媒体駆動装置
- 2 7 ネットワーク接続装置
- 2 8 バス
- 2 9 可搬記録媒体
- 3 0 プログラム（データ）提供者
- 3 1 回線
- S ステップ

【書類名】 図面

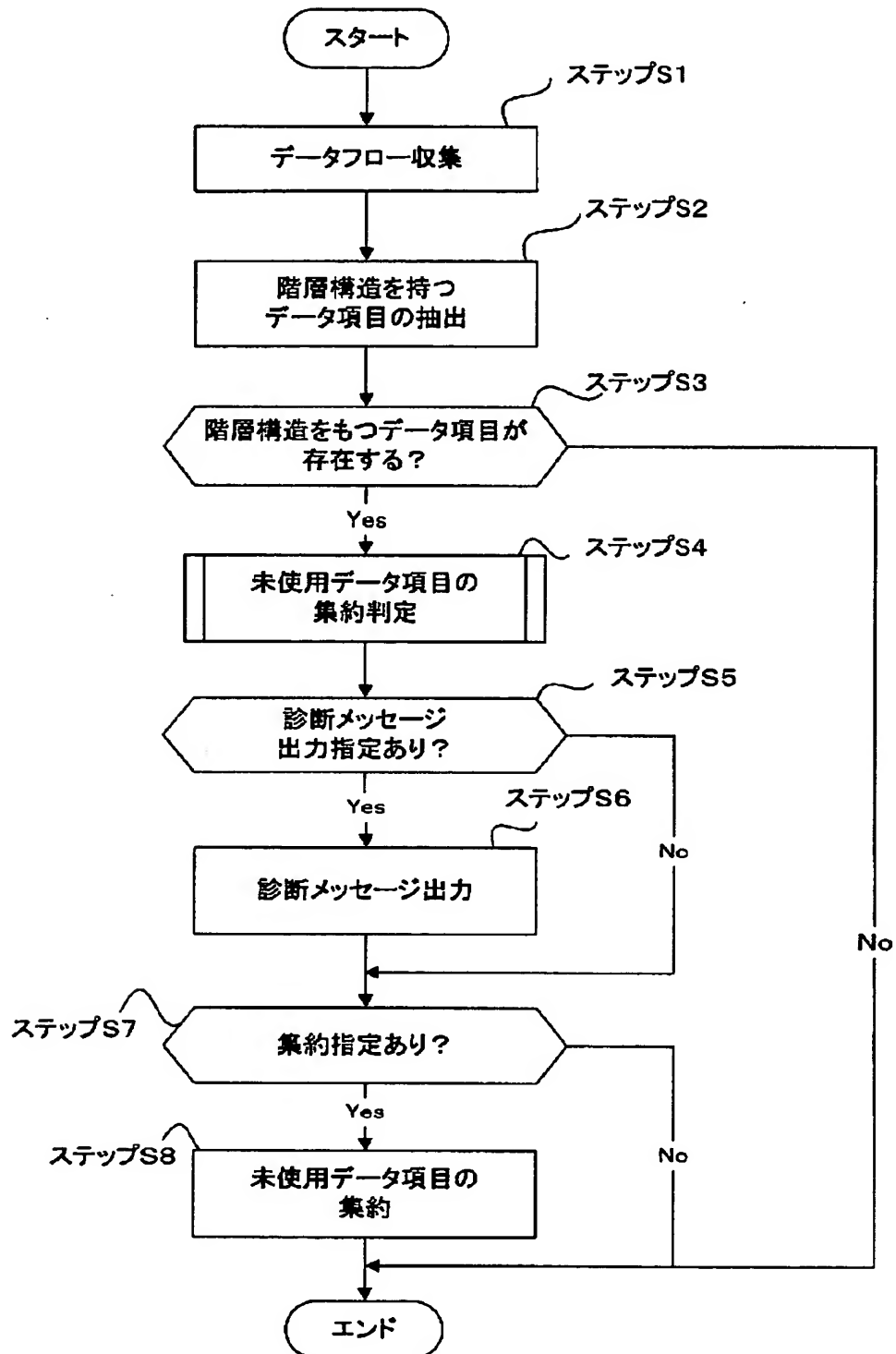
【図 1】

最適化処理装置の構成図



【図 2】

プログラムの最適化処理の大まかな流れを示すフローチャート



【図 3】

COBOLで記述されたソースプログラムの一例を示す図

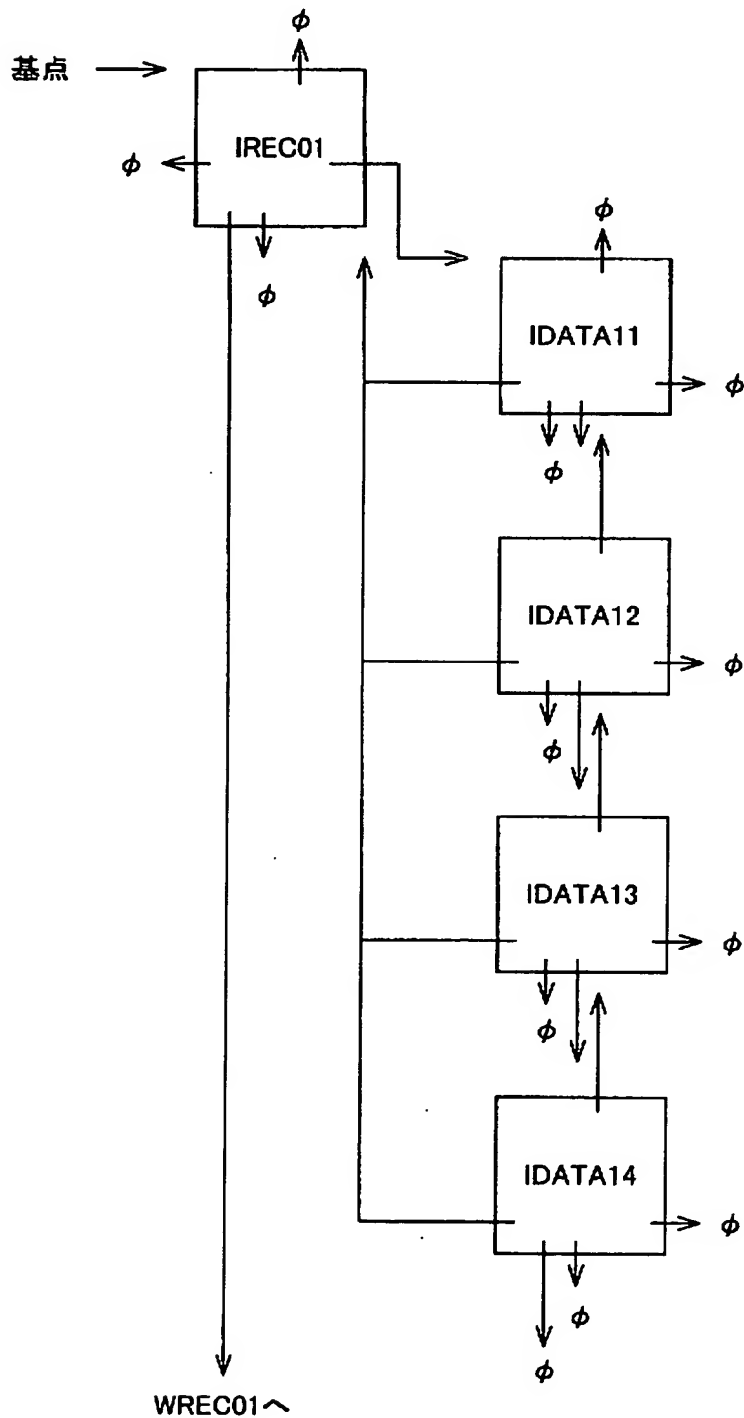
```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.    PRG1.
000030
000040 ENVIRONMENT    DIVISION.
000050 INPUT-OUTPUT    SECTION.
000060 FILE-CONTROL.
000070     SELECT IFILE01 ASSIGN SYSIN
000080 .
000090
000100 DATA            DIVISION.
000110 FILE              SECTION.
000120 FD IFILE01.
000130 01 IREC01.          *>未使用 ↑ 連続領域
000140 03 IDATA11 PIC X(20). *>未使用 |
000150 03 IDATA12 PIC X(20). *>未使用 |
000160 03 IDATA13 PIC X(20). *>未使用 |
000170 03 IDATA14 PIC X(20). *>未使用 ↓
000180 WORKING-STORAGE SECTION.
000190 01 WREC01.
000200 03 WDATA11 PIC 9(02).
000210 03 WDATA12 PIC 9(02). *>未使用 ↑ 連続領域
000220 03 WDATA13 PIC 9(02). *>未使用 ↓
000230 03 WDATA14 PIC 9(02).
000240 01 WREC02.          *>未使用 ↑ 連続領域
000250 03 WDATA21 PIC X(40). *>未使用 |
000260 03 WDATA22 PIC X(40). *>未使用 ↓
000270 77 WDATA71 PIC 9(02). *>未使用
000280 77 WDATA72 PIC 9(02).
000290
000300 PROCEDURE        DIVISION.
000310     MOVE 1 TO WDATA11 OF WREC01.
000320     MOVE 2 TO WDATA14 OF WREC01.
000330     COMPUTE WDATA72 = WDATA11 OF WREC01 + WDATA14 OF WREC01.
000340     DISPLAY WDATA72.
000350     STOP RUN.

```

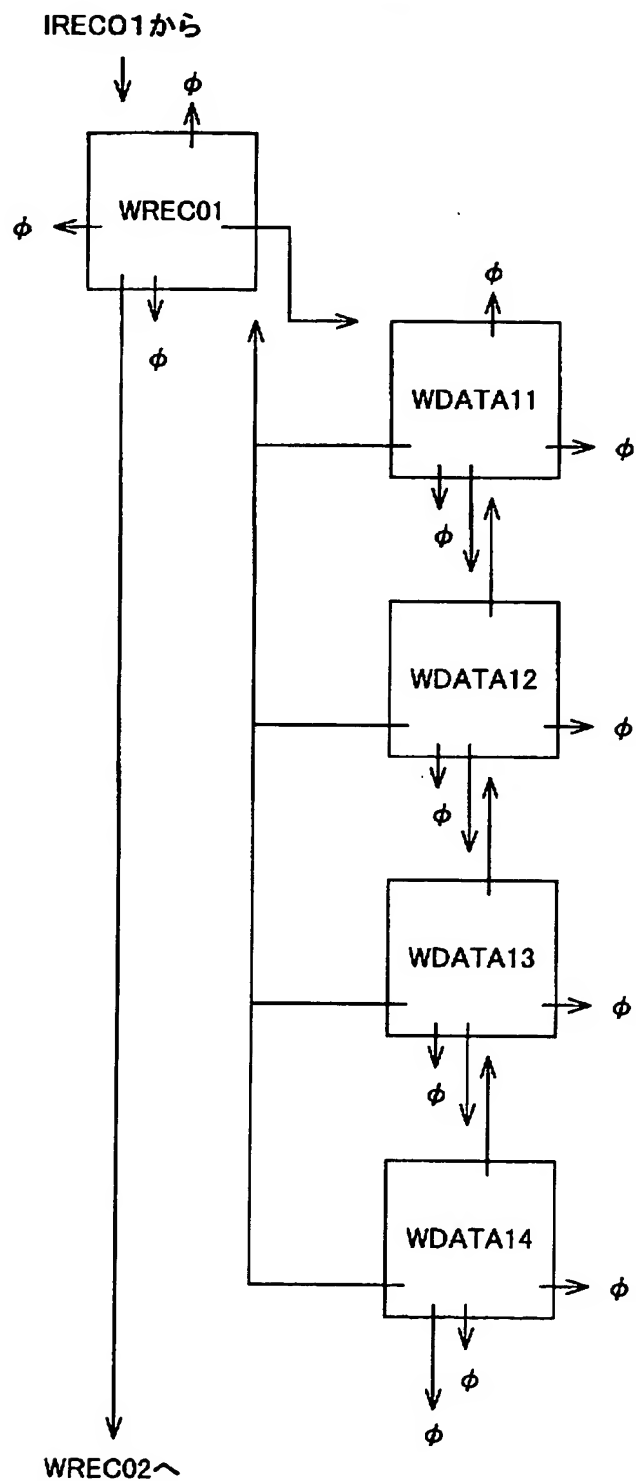
【図 4】

図3に示すソースプログラムの
データ項目辞書を示す図(その1)



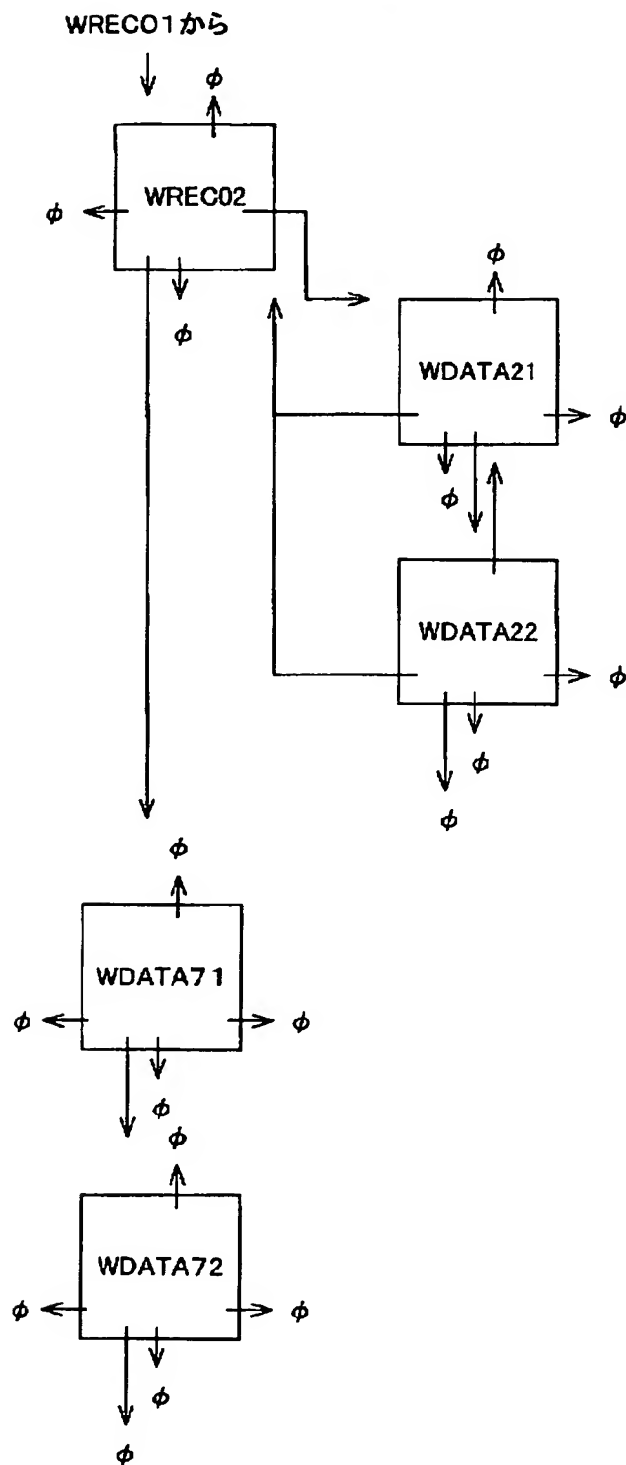
【図 5】

図3に示すソースプログラムの
データ項目辞書を示す図(その2)



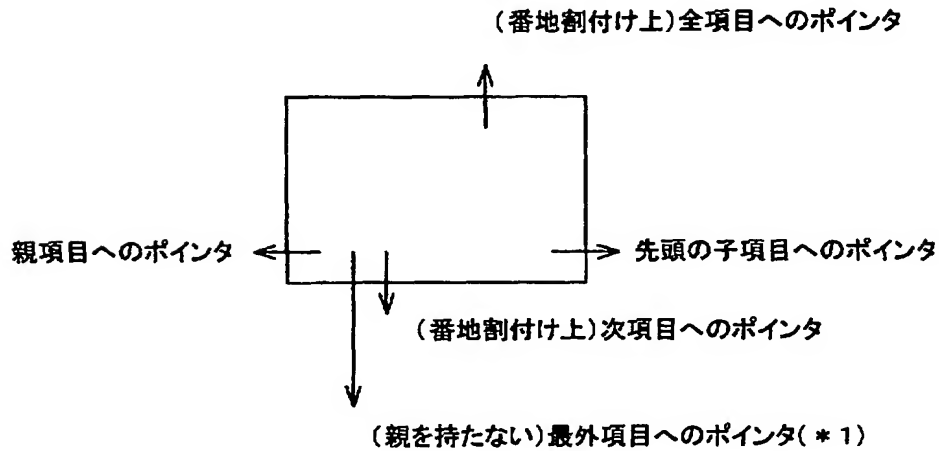
【図 6】

図3に示すソースプログラムの
データ項目辞書を示す図(その3)

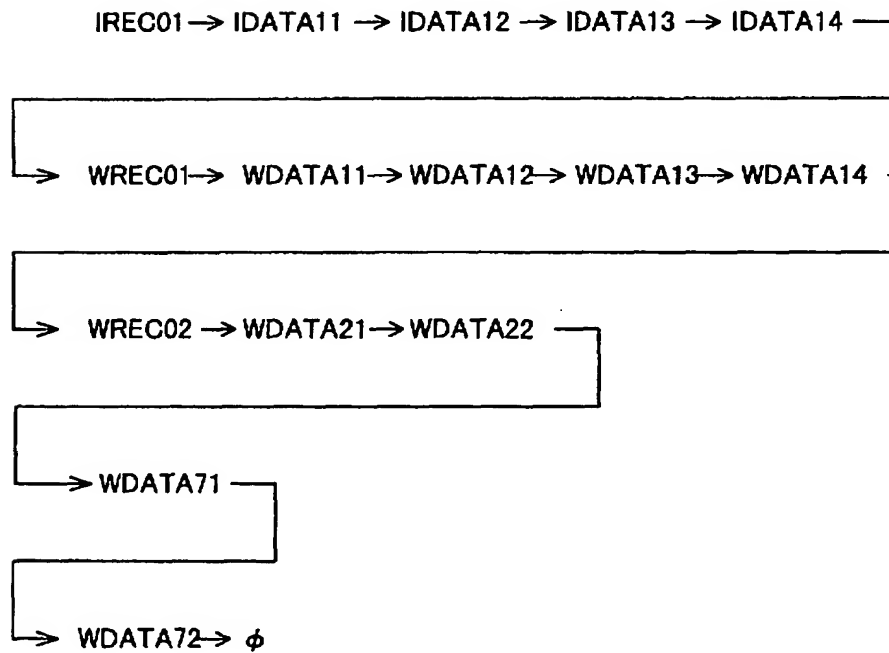


【図 7】

データ項目辞書の表記方法を説明する図



(a)



(b)

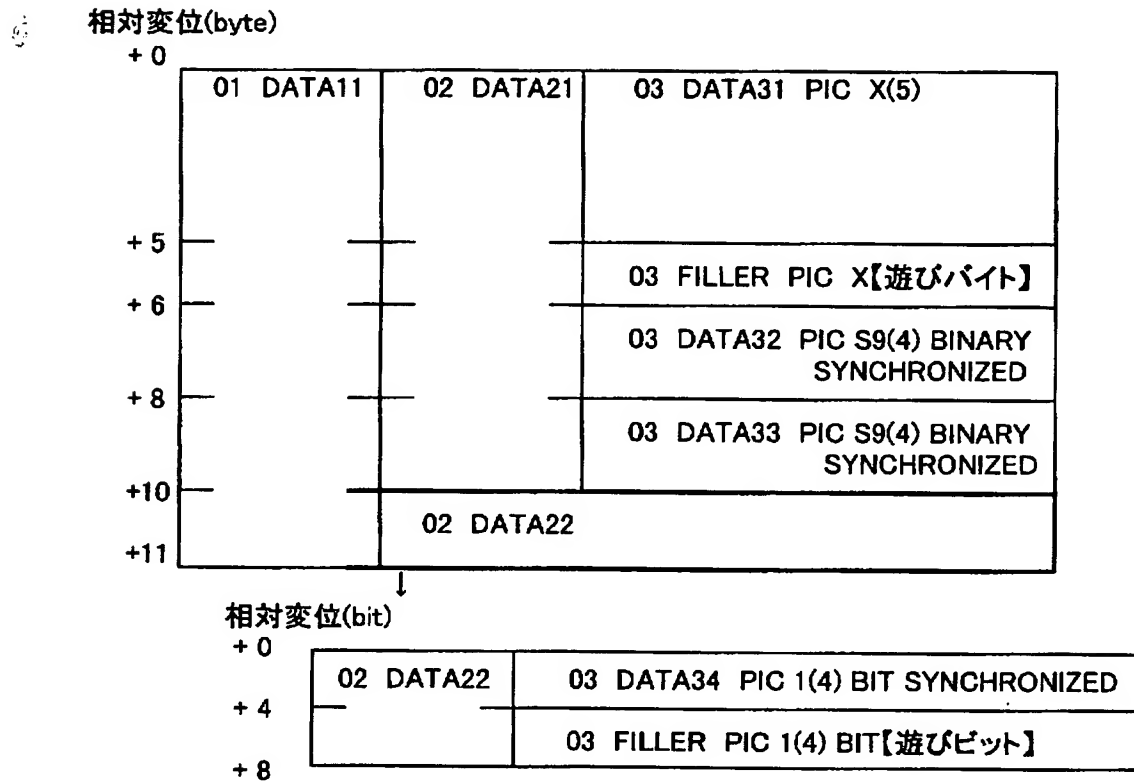
【図 8】

データ項目のレイアウトの一例を示す図(その1)

相対変位		
+ 0	01 IREC01	02 IDATA11 PIC X(20)
+20		02 IDATA12 PIC X(20)
+40		02 IDATA13 PIC X(20)
+60		02 IDATA14 PIC X(20)
+80		

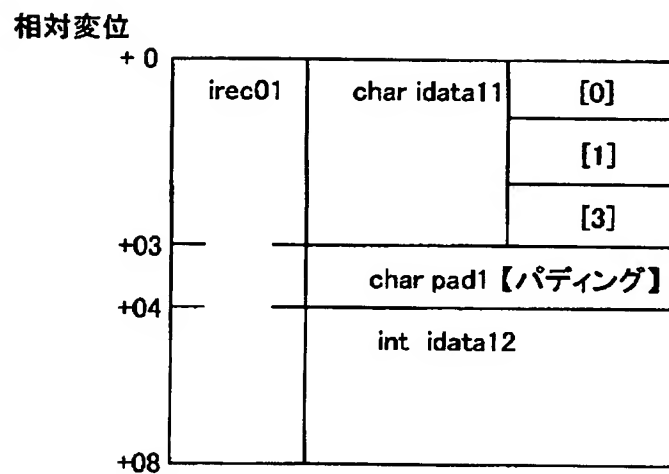
【図 9】

データ項目のレイアウトの一例を示す図(その2)



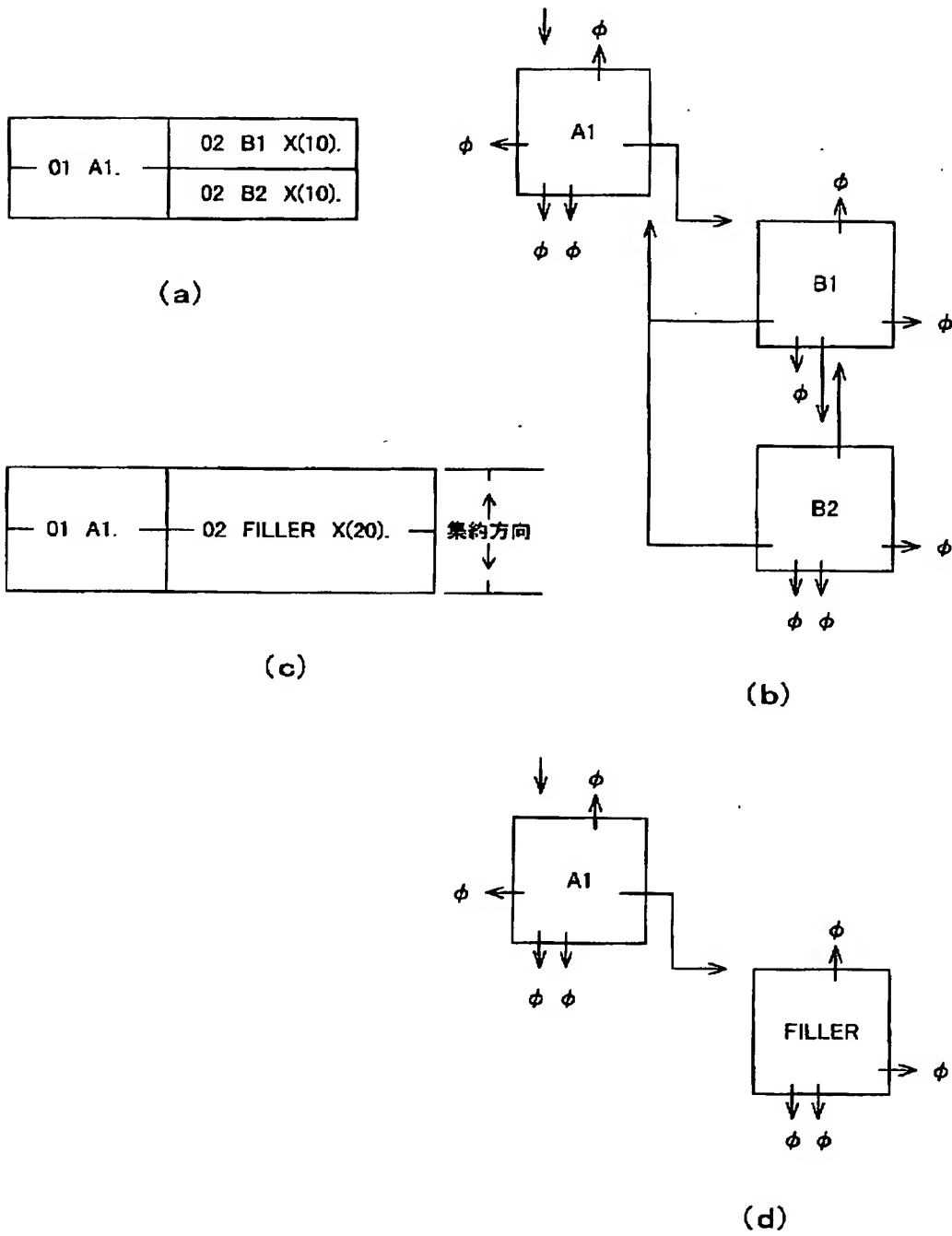
【図 1 0】

データ項目のレイアウトの一例を示す図(その3)



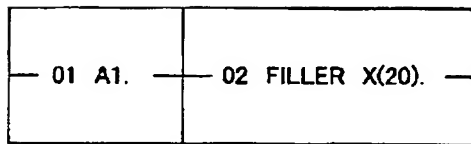
【図 1 1】

前後方向の集約を説明する図

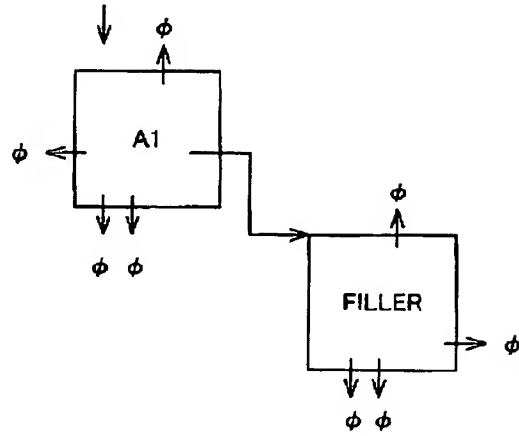


【図 1 2】

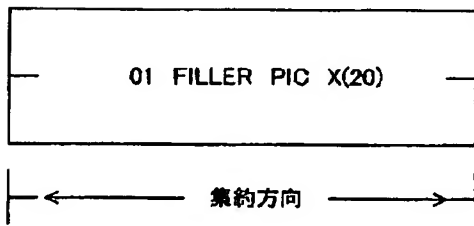
親子方向の集約を説明する図



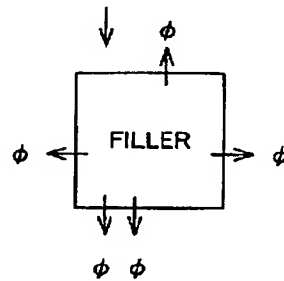
(a)



(b)



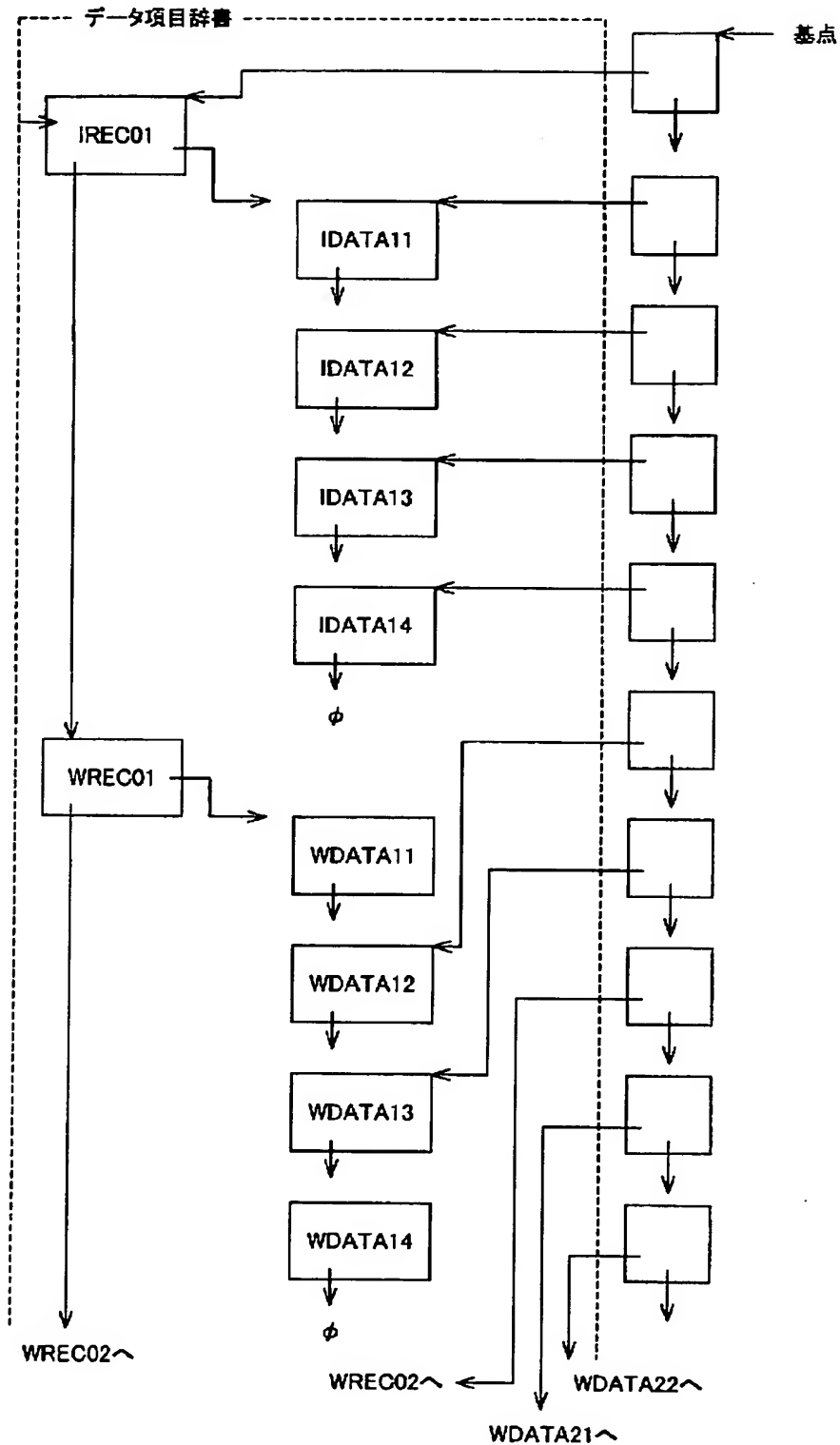
(c)



(d)

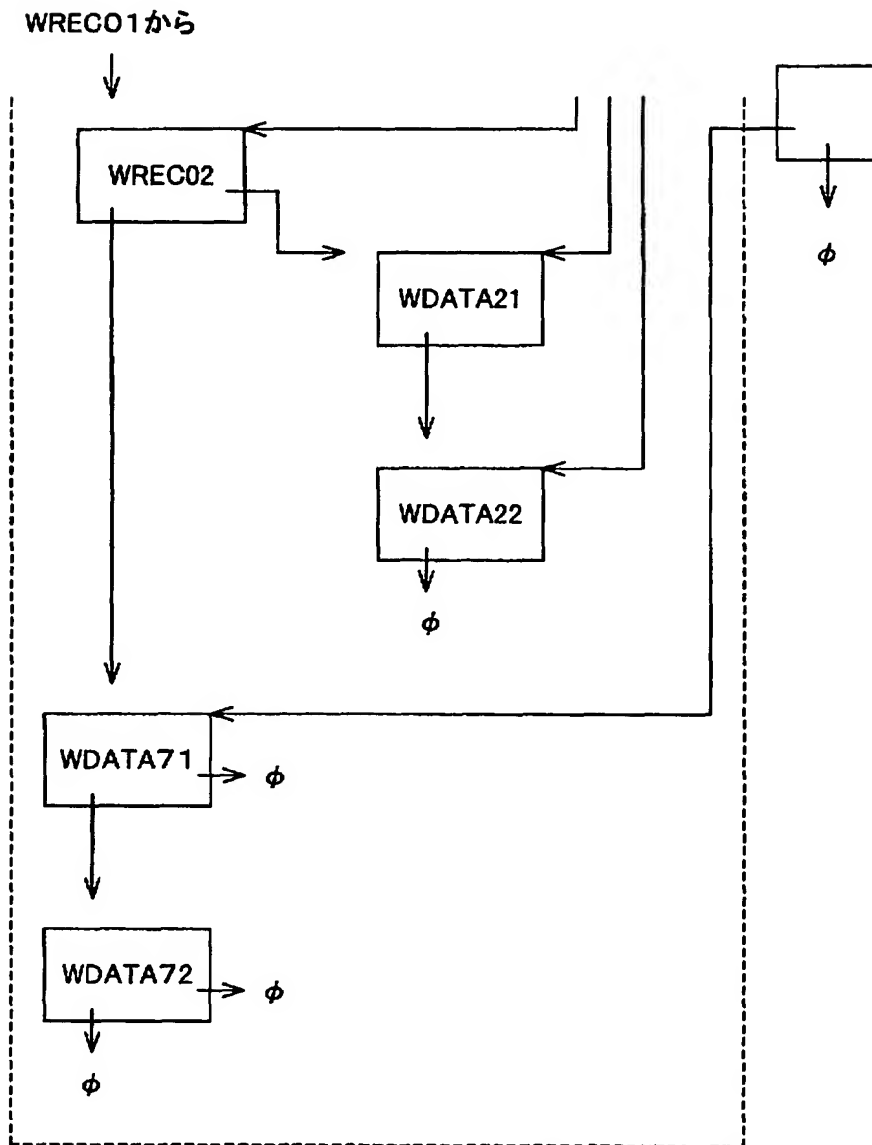
【図 13】

未判定データ項目リストに含まれる
ポインタを説明する図(その1)



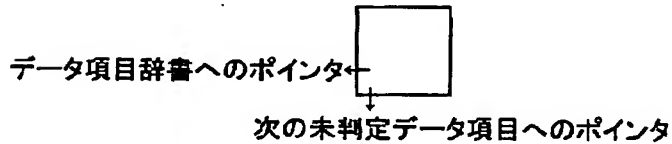
【図 1 4】

未判定データ項目リストに含まれる
ポインタを説明する図(その2)



【図 1 5】

図13及び図14に示す未判定データ項目リストのポインタの表記法
及び未判定データ項目リストに含まれる情報を示す図

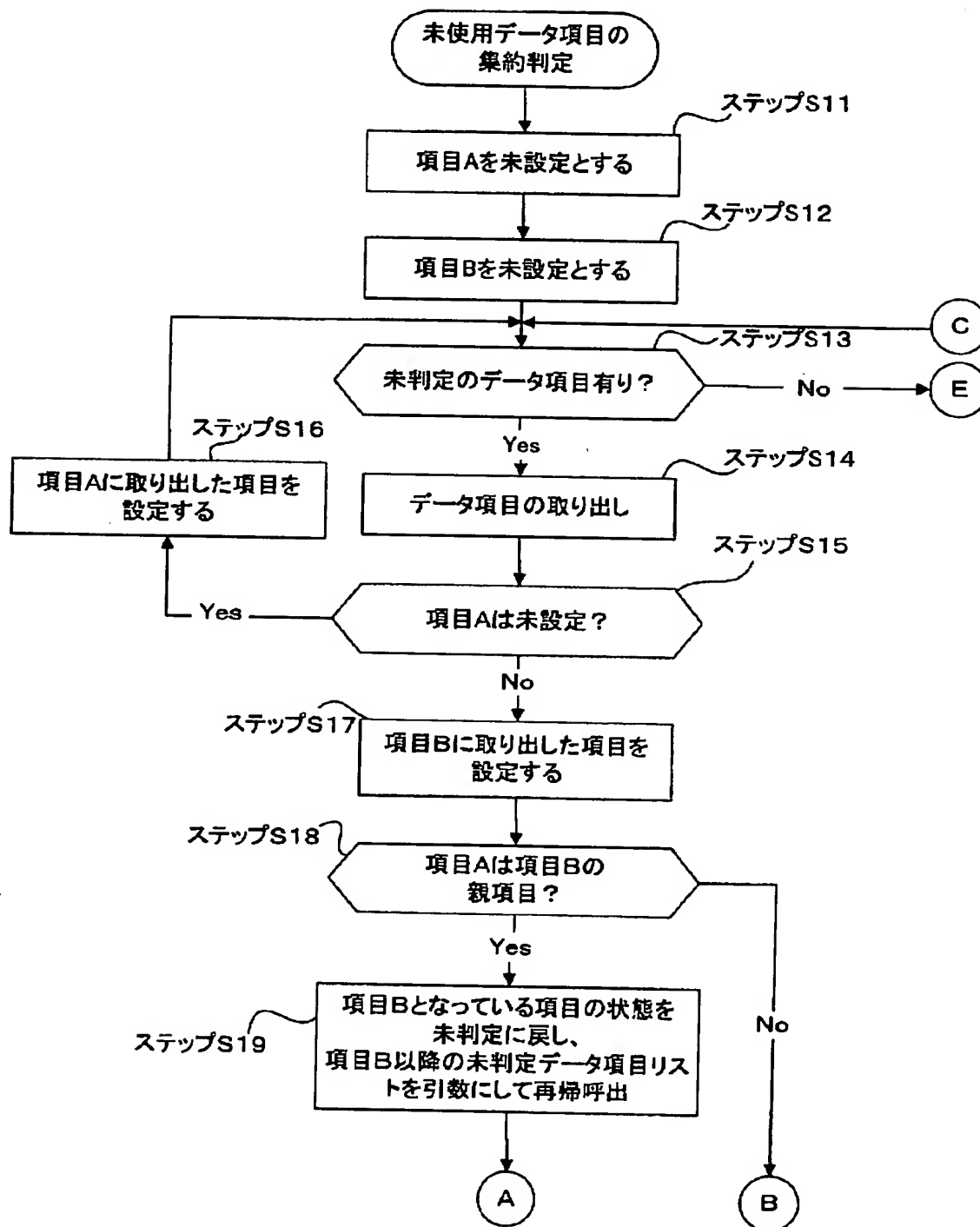


ポインタ以外の情報は以下の通り。データ項目群の集約判定の実行結果を保持する。

- 状態: { 未判定 | 判定済 }
 - 未判定: データ項目の集約判定を行なっていない
 - 判定済: データ項目の集約判定を行なった
- 結果: { 集約可 | 削除可 }
 - 集約可: 他のデータ項目との集約が可能(項目Aに相当)
 - 削除可: 他のデータ項目との集約が可能(項目Bに相当)
- 長さ: 「状態」が集約可能の場合、意味を持つ
集約後の(項目A)の長さ
- 型 : 「状態」が集約可能の場合、意味を持つ
集約後の(項目A)のデータ型

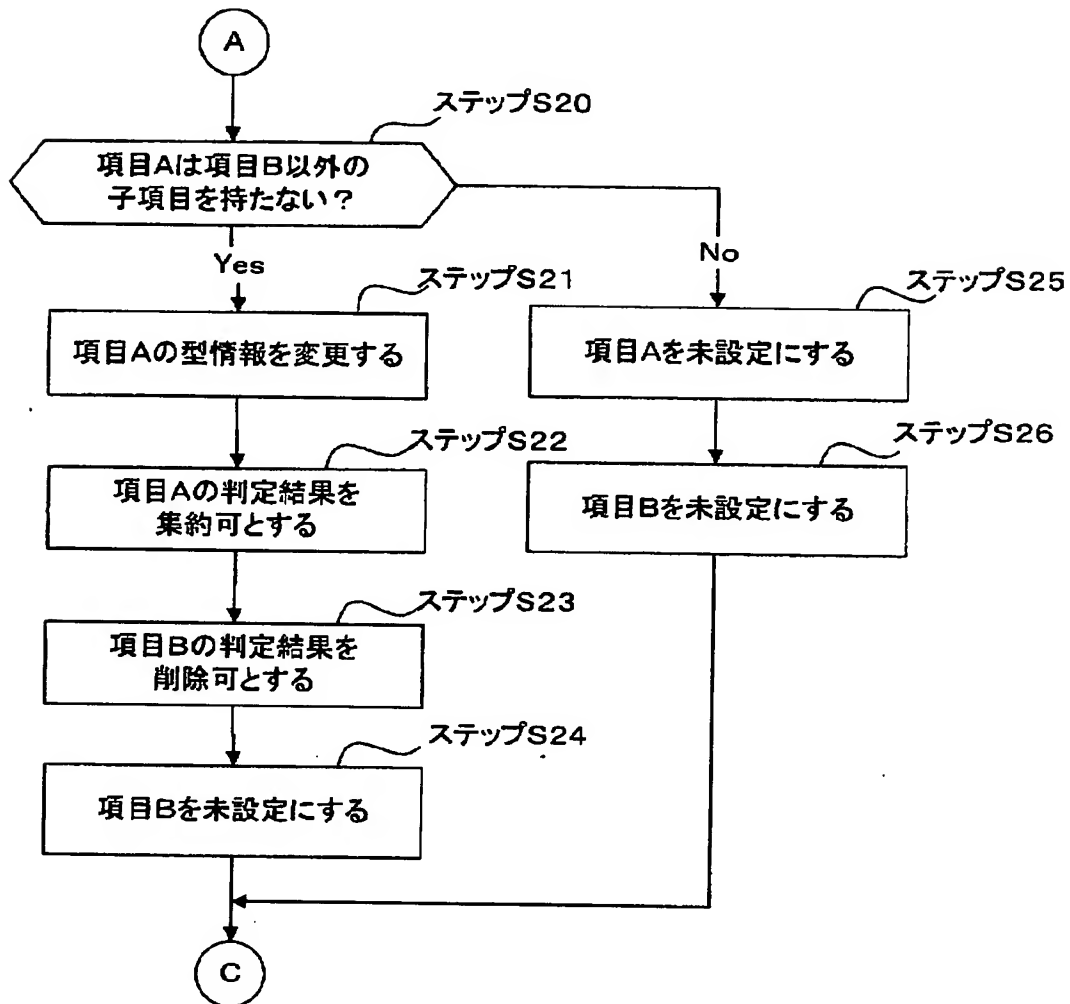
【図 16】

集約判定の手順を示すフローチャート(その1)



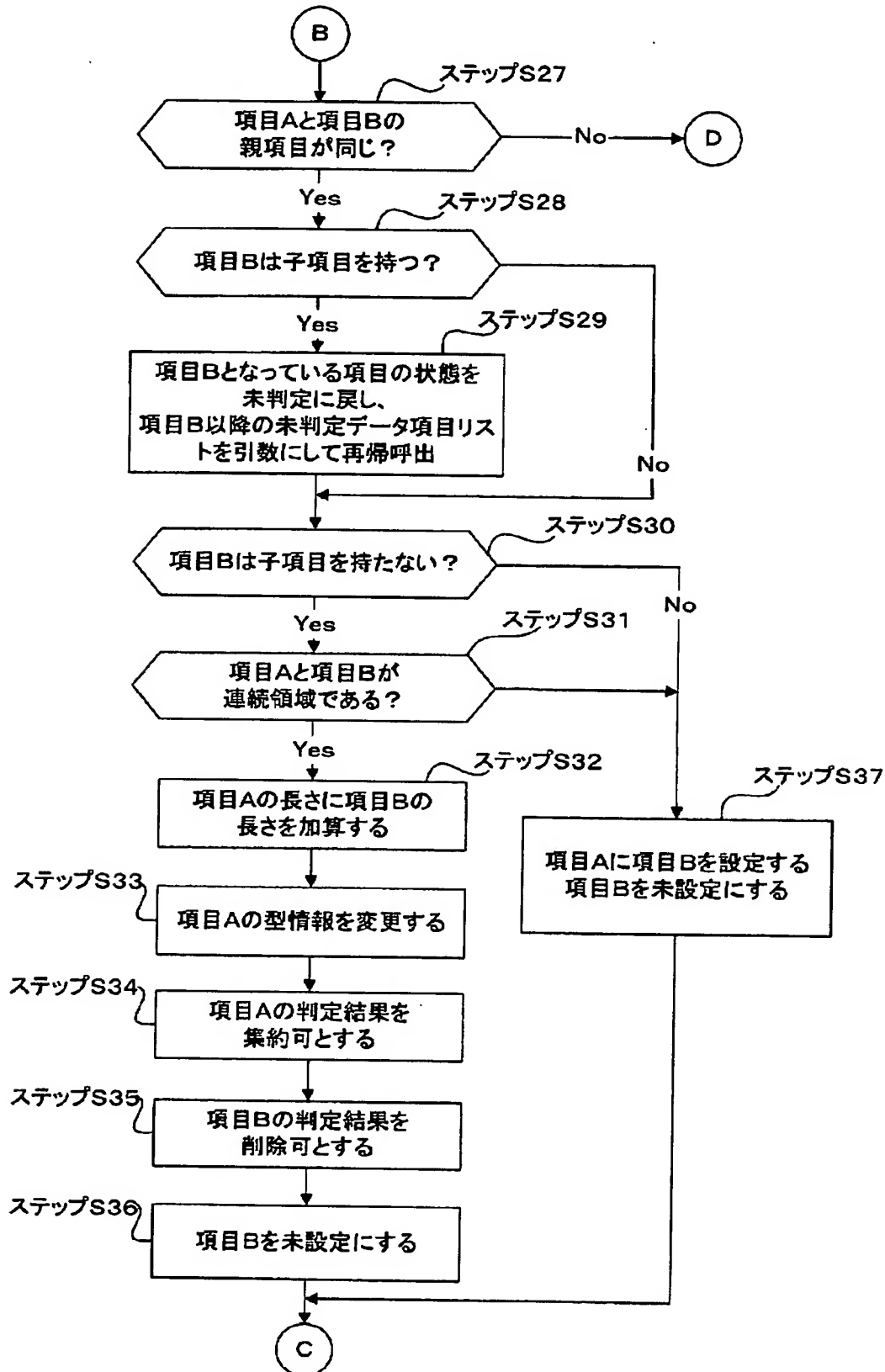
【図 1 7】

集約判定の手順を示すフローチャート(その2)



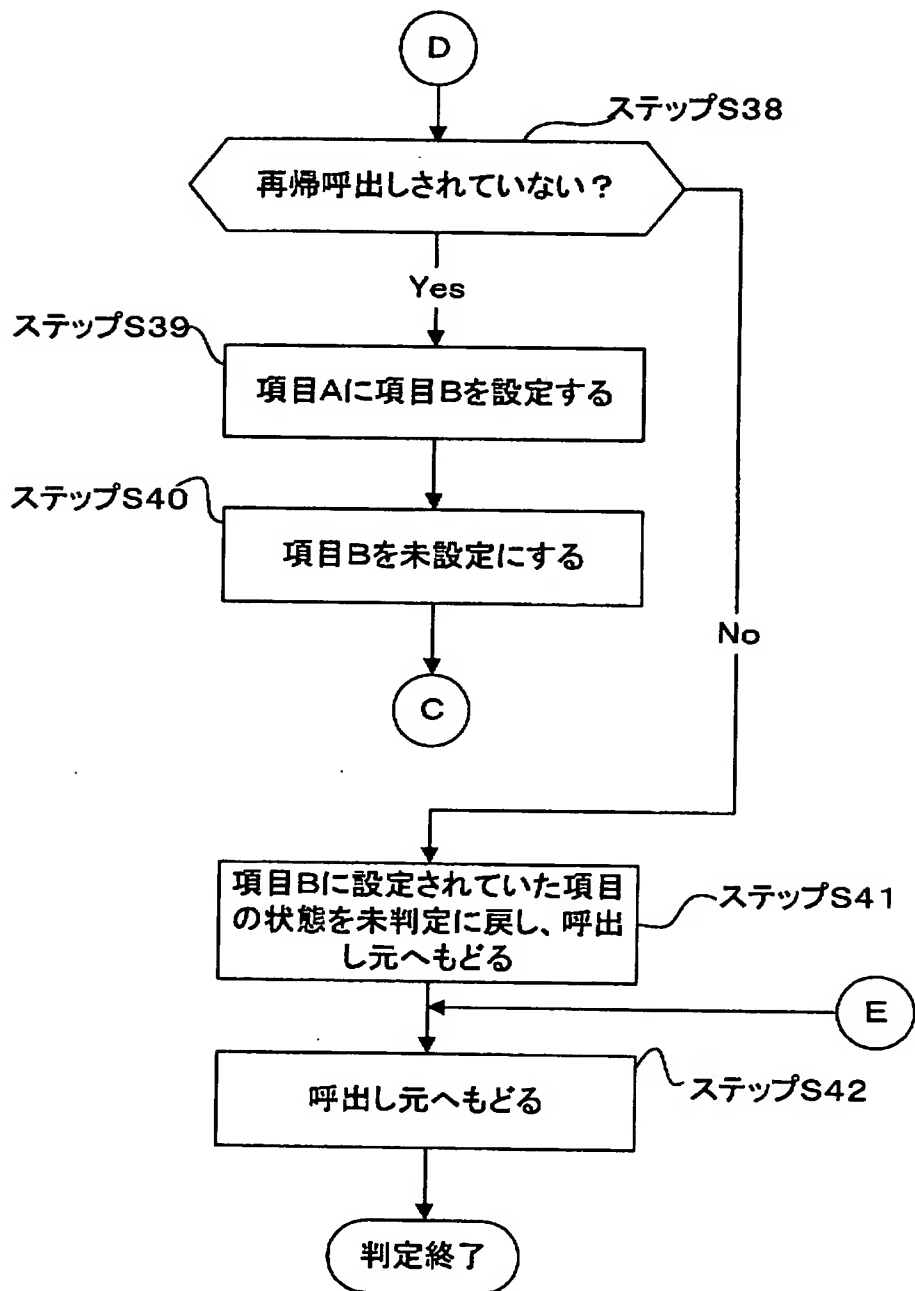
【図 1 8】

集約判定の手順を示すフローチャート(その3)



【図19】

集約判定の手順を示すフローチャート(その4)



【図 20】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その1)

段階	Step	一時格納域	項目A	項目B	未判定データ項目リスト
1	S00(0)	—	—	—	IREC01(未,—),IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
2	S11(0)	—	φ	—	IREC01(未,—),IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
3	S12(0)	—	φ	φ	IREC01(未,—),IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
4	S14(0)	IREC01	φ	φ	IREC01(済,—),IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
5	S16(0)	IREC01	IREC01	φ	IREC01(済,—),IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
6	S14(0)	IDATA11	IREC01	φ	IREC01(済,—),IDATA11(済,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
7	S17(0)	IDATA11	IREC01	IDATA11	IREC01(済,—),IDATA11(済,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)

【図 2 1】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その2)

【S19から再帰呼出し】					
8	S00(1)	—	—	—	IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
9	S11(1)	—	φ	—	IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
10	S12(1)	—	φ	φ	IDATA11(未,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
11	S14(1)	IDATA11	φ	φ	IDATA11(済,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
12	S16(1)	IDATA11	IDATA11	φ	IDATA11(済,—),IDATA12(未,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
13	S14(1)	IDATA12	IDATA11	φ	IDATA11(済,—),IDATA12(済,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)
14	S17(1)	IDATA12	IDATA11	IDATA12	IDATA11(済,—),IDATA12(済,—), IDATA13(未,—),IDATA14(未,—),WDATA12(未,—), WDATA13(未,—),WREC02(未,—),WDATA21(未,—), WDATA22(未,—)

【図 2 2】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その3)

15	S34(1)	IDATA12	IDATA11	IDATA12	IDATA11(済,集),IDATA12(済,-), IDATA13(未,-),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
16	S35(1)	IDATA12	IDATA11	IDATA12	IDATA11(済,集),IDATA12(済,削), IDATA13(未,-),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
17	S38(1)	IDATA12	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(未,-),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
18	S14(1)	IDATA13	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(済,-),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
19	S17(1)	IDATA13	IDATA11	IDATA13	IDATA11(済,集),IDATA12(済,削), IDATA13(済,-),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
20	S35(1)	IDATA13	IDATA11	IDATA13	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
21	S36(1)	IDATA13	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(未,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)

【図 23】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その4)

22	S14(1)	IDATA14	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
23	S17(1)	IDATA14	IDATA11	IDATA14	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,-),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
24	S35(1)	IDATA14	IDATA11	IDATA14	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
25	S36(1)	IDATA14	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
26	S14(1)	WDATA12	IDATA11	φ	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
27	S17(1)	WDATA12	IDATA11	WDATA12	IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)
【再帰呼出しから復帰】					
28	S19(0)	IDATA11	IREC01	IDATA11	IREC01(済,-),IDATA11(済,集),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(未,-), WDATA13(未,-),WREC02(未,-),WDATA21(未,-), WDATA22(未,-)

【図 24】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その5)

29	S22(0)	IDATA11	IREC01	IDATA11	IREC01 (済.集),IDATA11(済.集),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(未.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
30	S23(0)	IDATA11	IREC01	IDATA11	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(未.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
31	S24(0)	IDATA11	IREC01	φ	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(未.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
32	S14(0)	WDATA12	IREC01	φ	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(済.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
33	S17(0)	WDATA12	IREC01	WDATA12	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(済.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
34	S39(0)	WDATA12	WDATA12	WDATA12	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(済.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)
35	S40(0)	WDATA12	WDATA12	φ	IREC01 (済.集),IDATA11(済.削),IDATA12(済.削), IDATA13(済.削),IDATA14(済.削),WDATA12(済.-), WDATA13(未.-),WREC02 (未.-),WDATA21(未.-), WDATA22(未.-)

【図 25】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その6)

36	S14(0)	WDATA13	WDATA12	φ	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,一), WDATA13(済,一),WREC02 (未,一),WDATA21(未,一), WDATA22(未,一)
37	S17(0)	WDATA13	WDATA12	WDATA13	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,一), WDATA13(済,一),WREC02 (未,一),WDATA21(未,一), WDATA22(未,一)
38	S34(0)	WDATA13	WDATA12	WDATA13	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,一),WREC02 (未,一),WDATA21(未,一), WDATA22(未,一)
39	S35(0)	WDATA13	WDATA12	WDATA13	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (未,一),WDATA21(未,一), WDATA22(未,一)
40	S36(0)	WDATA13	WDATA12	φ	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (未,一),WDATA21(未,一), WDATA22(未,一)
41	S14(0)	WREC02	WDATA12	φ	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,一),WDATA21(未,一), WDATA22(未,一)
42	S17(0)	WREC02	WDATA12	WREC02	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,一),WDATA21(未,一), WDATA22(未,一)

【図 2 6】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その7)

43	S39(0)	WREC02	WREC02	WREC02	IREC01(済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02(済,一),WDATA21(未,一), WDATA22(未,一)
44	S40(0)	WREC02	WREC02	φ	IREC01(済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02(済,一),WDATA21(未,一), WDATA22(未,一)
45	S14(0)	WDATA21	WREC02	φ	IREC01(済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02(済,一),WDATA21(済,一), WDATA22(未,一)
46	S17(0)	WDATA21	WREC02	WDATA21	IREC01(済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02(済,一),WDATA21(済,一), WDATA22(未,一)
【S19から再帰呼出し】					
47	S00(1)	—	—	—	WDATA21(未,一),IDATA22(未,一)
48	S11(1)	—	φ	—	WDATA21(未,一),IDATA22(未,一)
49	S12(1)	—	φ	φ	WDATA21(未,一),IDATA22(未,一)
50	S14(1)	WDATA21	φ	φ	WDATA21(済,一),IDATA22(未,一)
51	S16(1)	WDATA21	WDATA21	φ	WDATA21(済,一),IDATA22(未,一)
52	S14(1)	WDATA22	WDATA21	φ	WDATA21(済,一),IDATA22(済,一)
53	S34(1)	WDATA22	WDATA21	WDATA22	WDATA21(済,集),IDATA22(済,一)
54	S35(1)	WDATA22	WDATA21	WDATA22	WDATA21(済,集),IDATA22(済,削)

【図 27】

図3に示すソースプログラムについて集約判定を行う場合の各段階における項目A
及び項目Bの設定内容並びに未判定データ項目リストの内容を示す図(その8)

55	S36(1)	WDATA22	WDATA21	φ	WDATA21(済,集),IDATA22(済,削)
【再帰呼出しから復帰】					
56	S19(0)	WDATA21	WREC02	WDATA21	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,一),WDATA21(済,集), WDATA22(済,削)
57	S22(0)	WDATA21	WREC02	WDATA21	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,集),WDATA21(済,集), WDATA22(済,削)
58	S23(0)	WDATA21	WREC02	WDATA21	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,集),WDATA21(済,削), WDATA22(済,削)
59	S24(0)	WDATA21	WREC02	φ	IREC01 (済,集),IDATA11(済,削),IDATA12(済,削), IDATA13(済,削),IDATA14(済,削),WDATA12(済,集), WDATA13(済,削),WREC02 (済,集),WDATA21(済,削), WDATA22(済,削)

【図 2 8】

図3に示すソースプログラムを最適化した結果を示す図

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. PRG1.
000030
000040 ENVIRONMENT DIVISION.
000050 INPUT-OUTPUT SECTION.
000060 FILE-CONTROL.
000070 SELECT IFILE01 ASSIGN SYSIN
000080 .
000090
000100 DATA DIVISION.
000110 FILE SECTION.
000120 FD IFILE01.
000130 01 FILLER PIC X(80). *>未使用、集約後の変数
000140* 集約↑
000150* 集約↑
000160* 集約↑
000170* 集約↑
000180 WORKING-STORAGE SECTION.
000190 01 WREC01.
000200 03 WDATA11 PIC 9(02).
000210 03 FILLER PIC 9(04). *>未使用、集約後の変数
000220* 集約↑
000230 03 WDATA14 PIC 9(02).
000240 01 FILLER PIC X(80). *>未使用、集約後の変数
000250* 集約↑
000260* 集約↑
000270 77 WDATA71 PIC 9(02). *>未使用
000280 77 WDATA72 PIC 9(02).
000290
000300 PROCEDURE DIVISION.
000310 MOVE 1 TO WDATA11 OF WREC01.
000320 MOVE 2 TO WDATA14 OF WREC01.
000330 COMPUTE WDATA72 = WDATA11 OF WREC01 + WDATA14 OF WREC01.
000340 DISPLAY WDATA72.
000350 STOP RUN.

```

【図 2 9】

図28に示す最適化されたソースプログラムから
不要な定義コードを削除した結果を示す図

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.    PRG1.
000030
000040 ENVIRONMENT    DIVISION.
000050 INPUT-OUTPUT    SECTION.
000060 FILE-CONTROL.
000070     SELECT IFILE01 ASSIGN SYSIN
000080
000090
000100 DATA          DIVISION.
000110 FILE            SECTION.
000120 FD IFILE01.
000130 01 FILLER     PIC X(80). *>未使用、集約後の変数
000140* 集約 ↑
000150* 集約 ↑
000160* 集約 ↑
000170* 集約 ↑
000180 WORKING-STORAGE SECTION.
000190 01 WREC01.
000200 03 WDATA11 PIC 9(02).
000210 03 FILLER PIC 9(04). *>未使用、集約後の変数
000220* 集約 ↑
000230 03 WDATA14 PIC 9(02).
000240* 削除
000250* 集約 ↑
000260* 集約 ↑
000270* 削除
000280 77 WDATA72 PIC 9(02).
000290
000300 PROCEDURE      DIVISION.
000310     MOVE 1 TO WDATA11 OF WREC01.
000320     MOVE 2 TO WDATA14 OF WREC01.
000330     COMPUTE WDATA72 = WDATA11 OF WREC01 + WDATA14 OF WREC01.
000340     DISPLAY WDATA72.
000350     STOP RUN.

```

【図 3 0】

Cで記述されたソースプログラムの一例を示す図

```

#include <stdio.h>

struct {
    char idata11[20]; /*未使用*/
    char idata12[20]; /*未使用*/
    char idata13[20]; /*未使用*/
    char idata14[20]; /*未使用*/
} irec01;          /*未使用*/

```

↑
連続領域
↓

```

void main(void) {
    struct {
        short wdata11;
        short wdata12; /*未使用*/
        short wdata13; /*未使用*/
        short wdata14;
    } wrec01;

    struct {
        char wdata21[40]; /*未使用*/
        char wdata22[40]; /*未使用*/
    } wrec02;          /*未使用*/

    short wdata71;      /*未使用*/
    short wdata72;

    wrec01.wdata11 = 1;
    wrec01.wdata14 = 2;
    wdata72 = wrec01.wdata11 + wrec01.wdata14;
    printf("%d\n", wdata72);
}

```

↑
連続領域
↓

↑
連続領域
↓

【図 3 1】

図30に示すソースプログラムを最適化した結果を示す図

```
#include<stdio.h>

char irec01[80]:    /*未使用、集約後の変数*/

void main(void) {
    struct{
        short wdata11;
        char wdata12[4]: /*未使用、集約後の変数*/
        short wdata14;
    }wrec01;

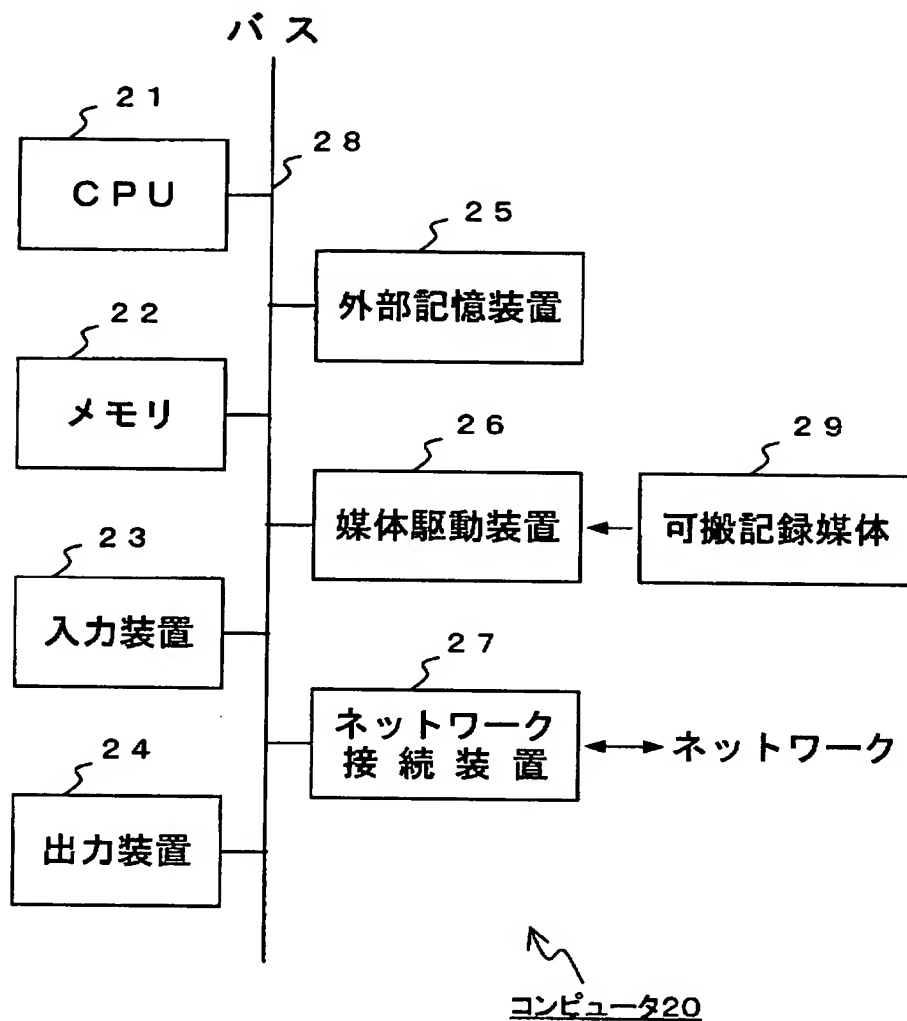
    char wrec02[80]:    /*未使用、集約後の変数*/

    short wdata71:      /*未使用*/
    short wdata72;

    wrec01.wdata11 = 1;
    wrec01.wdata14 = 2;
    wdata72=wrec01.wdata11 + wrec01.wdata14;
    printf("%d\n",wdata72);
}
```

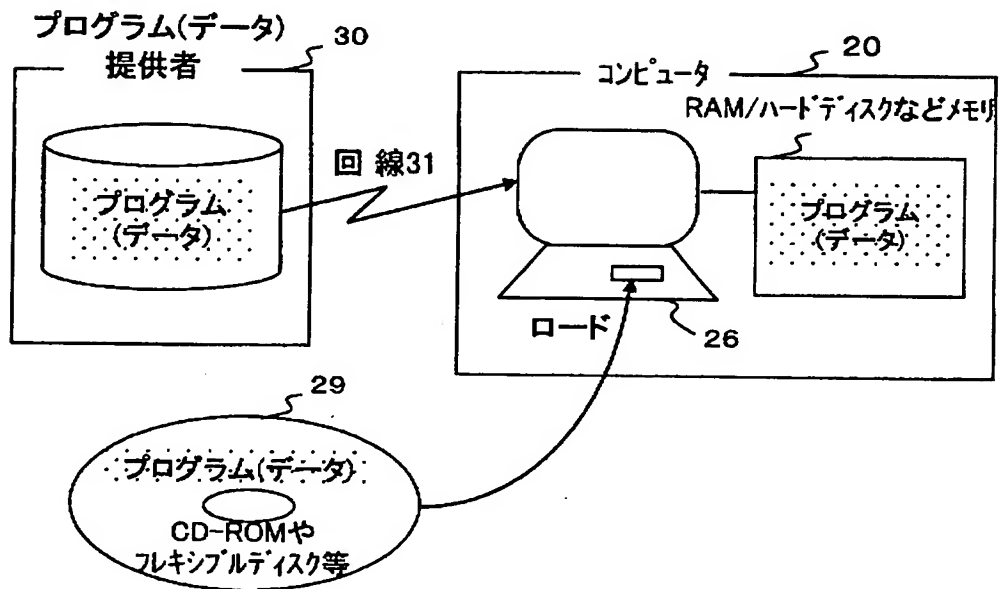
【図 32】

コ ン ピ ュ ー タ の 構 成 図



【図 33】

プログラムやデータのコンピュータへのローディングを説明する図



【図 3 4】

COBOLで記述されたソースプログラムの一例を示す図

```
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. PRG1.
000030
000040 ENVIRONMENT DIVISION.
000050 INPUT-OUTPUT SECTION.
000060 FILE-CONTROL.
000070 SELECT IFILE01 ASSIGN SYSIN
000080 .
000090
000100 DATA DIVISION.
000110 FILE SECTION.
000120 FD IFILE01.
000130 01 IREC01. *>未使用
000140 03 IDATA11 PIC X(20). *>未使用
000150 03 IDATA12 PIC X(20). *>未使用
000160 03 IDATA13 PIC X(20). *>未使用
000170 03 IDATA14 PIC X(20). *>未使用
000180 WORKING-STORAGE SECTION.
000190 01 WREC01.
000200 03 WDATA11 PIC 9(02).
000210 03 WDATA12 PIC 9(02). *>未使用
000220 03 WDATA13 PIC 9(02). *>未使用
000230 03 WDATA14 PIC 9(02).
000240 01 WREC02. *>未使用
000250 03 WDATA21 PIC X(40). *>未使用
000260 03 WDATA22 PIC X(40). *>未使用
000270 77 WDATA71 PIC 9(02). *>未使用
000280 77 WDATA72 PIC 9(02).
000290
000300 PROCEDURE DIVISION.
000310 MOVE 1 TO WDATA11 OF WREC01.
000320 MOVE 2 TO WDATA14 OF WREC01.
000330 COMPUTE WDATA72 = WDATA11 OF WREC01 + WDATA14 OF WREC01.
000340 DISPLAY WDATA72.
000350 STOP RUN.
```


【図 3' 5】

図34に示すソースプログラムから従来技術を用いて
未使用データ項目を定義するコードを削除した結果を示す図

```

000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID.    PRG1.
000030
000040 ENVIRONMENT    DIVISION.
000050 INPUT-OUTPUT    SECTION.
000060 FILE-CONTROL.
000070    SELECT IFILE01 ASSIGN SYSIN
000080 .
000090
000100 DATA          DIVISION.
000110 FILE            SECTION.
000120 FD IFILE01.
000130 01 IREC01.      *>未使用
000140 03 IDATA11 PIC X(20). *>未使用
000150 03 IDATA12 PIC X(20). *>未使用
000160 03 IDATA13 PIC X(20). *>未使用
000170 03 IDATA14 PIC X(20). *>未使用
000180 WORKING-STORAGE SECTION.
000190 01 WREC01.
000200 03 WDATA11 PIC 9(02).
000210 03 WDATA12 PIC 9(02). *>未使用
000220 03 WDATA13 PIC 9(02). *>未使用
000230 03 WDATA14 PIC 9(02).
000240* 削除
000250* 削除
000260* 削除
000270* 削除
000280 77 WDATA72    PIC 9(02).
000290
000300 PROCEDURE      DIVISION.
000310    MOVE 1 TO WDATA11 OF WREC01.
000320    MOVE 2 TO WDATA14 OF WREC01.
000330    COMPUTE WDATA72 = WDATA11 OF WREC01 + WDATA14 OF WREC01.
000340    DISPLAY WDATA72.
000350    STOP RUN.

```

【図 3 6】

Cで記述されたソースプログラムの一例を示す図

```
#include <stdio.h>

struct {
    char idata11[20]; /*未使用*/
    char idata12[20]; /*未使用*/
    char idata13[20]; /*未使用*/
    char idata14[20]; /*未使用*/
} irec01; /*未使用*/

void main(void) {
    struct {
        short wdata11;
        short wdata12; /*未使用*/
        short wdata13; /*未使用*/
        short wdata14;
    } wrec01;

    struct {
        char wdata21[40]; /*未使用*/
        char wdata22[40]; /*未使用*/
    } wrec02; /*未使用*/

    short wdata71; /*未使用*/
    short wdata72;

    wrec01.wdata11 = 1;
    wrec01.wdata14 = 2;
    wdata72 = wrec01.wdata11 + wrec01.wdata14;
    printf("%d\n", wdata72);
}
```

【図 3 7】

図36に示すソースプログラムから従来技術を用いて
未使用データ項目を定義するコードを削除した結果を示す図

```
#include <stdio.h>

struct {
    char idata11[20]; /*削除*/
    char idata12[20]; /*削除*/
    char idata13[20]; /*削除*/
    char idata14[20]; /*削除*/
} irec01; /*削除*/

void main(void) {
    struct {
        short wdata11;
        short wdata12; /*未使用*/
        short wdata13; /*未使用*/
        short wdata14;
    } wrec01;

    /* 削除 */
    /* 削除 */
    /* 削除 */
    /* 削除 */

    /* 削除 */
    short wdata72;

    wrec01.wdata11 = 1;
    wrec01.wdata14 = 2;
    wdata72 = wrec01.wdata11 + wrec01.wdata14;
    printf("%d\n", wdata72);
}
```

【書類名】 要約書

【要約】

【課題】 効果的に未使用データ項目の数を削減することを可能とする。

【解決手段】 プログラムを最適化する最適化処理装置 1 に、データ項目抽出部 2、レイアウト部 4、未使用データ項目抽出部 5、集約判定部 6 及びデータ項目集約部 9 を備える。データ項目抽出部 2 は、プログラムからデータ項目を抽出する。レイアウト部 4 は、抽出されたデータ項目をメモリ上にレイアウトする。未使用データ項目抽出部 5 は、抽出されたデータ項目から、定義されたが使用されない未使用データ項目を抽出する。集約判定部 6 は、レイアウト結果に基づいて、未使用データ項目のうち、階層構造をもつデータ項目を構成する複数の未使用データ項目を、1 つの新たなデータ項目に集約することが可能か否か判定する。データ項目集約部 9 は、判定結果に基づいて、複数のデータ項目を 1 つのデータ項目に集約する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日
[変更理由] 住所変更
住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社